

Mobiilirobotti osana robottien internetiä

MiR100:n ohjaus ulkoisella järjestelmällä

Alexi Nuora

Opinnäytetyö

Toukokuu 2020

Tekniikan ala

Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Automaatiotekniikka

Tekijä(t) Nuora, Aleksi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2020
	Sivumäärä 51	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi Mobiilirobotti osana robottien internetiä MiR100:n ohjaus ulkoisella järjestelmällä		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikka, insinööri (AMK)		
Työn ohjaaja(t) Juho Riekkinen, Markku Ström		
Toimeksiantaja(t) Transval Group Oy		
Tiivistelmä Mobiilirobotti MiR100 liitettiin osaksi robottien internet eli IoRT-järjestelmää. Robotin ohjaukseen ulkoisesta järjestelmästä luotiin toimintamalli, jonka avulla robotti voitiin kutsua haluttuun sijaintiin. Sijainnin paikkakoordinaatit eivät olleet mobiilirobotin käyttöjärjestelmässä ennalta määrättyjä vaan ne lähetettiin ulkoisesta järjestelmästä robotin kutsun yhteydessä. Käytetyn IoRT-järjestelmän lisäksi mobiilirobotin ohjaus ulkoisesta järjestelmästä osoitettiin toimivaksi, käyttäen robotin kanssa samassa paikallisverkossa toiminutta PC:ä sekä Raspberry Pi –korttitietokonetta. Pyrkimyksenä oli hyödyntää toteutettua ohjaustapaa mobiilirobotisovelluksessa, jossa robotti vie täyden roska-astian tyhjennykseen, astiaan kiinnitetyn anturin IoRT-järjestelmälle lähettämän tilatiedon perusteella. Maailmanlaajuinen kriisi vaikeutti merkittävästi roska-astiasovelluksen käytännön testausten toteutusta. Robotilla toteutetun ohjaustavan myötä mahdollistui kuitenkin ideoida robotin kutsusovellus, joka olisi markkinoilla nykyisin olevia kutsunappisovelluksia selkeästi edullisempi. Tavoitteen taustalla oli myös jatkokehitysidea vastaavan sovelluksen automatisoinnista. Mobiilirobotin IoRT-pohjaisen ohjauksen automatisoidussa versiossa robotin kutsusijainti määräytyisi dynaamisesti vasta kutsuhetkellä. Tällainen sovellus mahdollistaisi sisälogistiikan materiaalinsiirtoihin liittyvien työvaiheiden automaatioasteen merkittävän kasvattamisen. Samalla syntyisi täysin uudenlaisia mobiilirobotiikan käyttöönottomahdollisuuksia myös muilla teollisuuden toimialoilla. Mahdollisuudet toteuttaa vastaava automatisoitu sovellus tutkittiin selvittämällä markkinoilla olevien sisäpaikannusjärjestelmien paikoitustarkkuus. Vaihtoehtoisena toteutuksena ideoitui jatkokehitysmalli, jossa mobiilirobotin kutsusijainti määräytyisi dynaamisesti konenäkösovelluksen avulla. Tällainen teollisuusnivelesiirroboteille tuttu toimintamalli olisi uusi mobiilirobotiikassa.		
Avainsanat (asiasanat) robotit, robotiikka, teollisuusautomaatio, esineiden internet, asioiden internet, IoT, mobiilirobotit, mobiilirobotiikka, robottien internet, IoRT, Raspberry Pi, REST API		
Muut tiedot (salassa pidettävät liitteet) Ei salassa pidettäviä liitteitä		

Description

Author(s) Nuora, Aleks	Type of publication Bachelor's thesis	Date May 2020
	Number of pages 51	Language of publication: Finnish
		Permission for web publication: x
Title of publication Mobile Robot as a Part of Internet of Robotic Things MiR100's Control with External System		
Degree programme Electrical and automation engineering, Engineer		
Supervisor(s) Riekkinen Juho, Ström Markku		
Assigned by Transval Group Oy		
<p>Abstract</p> <p>Mobile robot MiR100 was connected to an Internet of Robotic Things i.e. IoRT system. For controlling the robot from an external system a proof of concept was created with which the robot could be called to a wanted position. Coordinates of this position weren't predetermined in the robot's operating system but they were sent from the external system with the call of the robot. In addition to the IoRT system used, controlling the mobile robot with an external system was proven to work using a PC and a Raspberry Pi single-board computer in the same local network. The objective was to utilize the controlling procedure achieved in a mobile robot application in which the robot delivers full rubbish bin to dumping initiated by state information sent to the IoRT system by a sensor attached to the bin. Global crisis hindered substantially the practical testing of the rubbish bin application. However, due to the controlling procedure implemented for the mobile robot it was possible to think up an application for calling the robot which would be significantly cheaper than the current applications for the task at the market today. As an additional objective was an idea to automate the external controlling of the robot. In an automated version of the IoRT-based mobile robot system the robot's target position would be determined at the moment of the robot call. This kind of application would enable a substantial increase in the degree of automation of working phases of material transfers in internal logistics. At the same time, whole new possibilities to utilize mobile robotics would emerge for other fields of industry also. Means to implement such an application were researched by surveying the locating precision of indoor position systems at the market today. As an alternative, a further development model was thought up in which the target position of the mobile robot would be determined dynamically with a machine vision application. This would be a new operations model for mobile robots.</p>		
<p>Keywords (subjects)</p> <p>robots, robotics, industrial automation, internet of things, IoT, mobile robot, mobile robotics, internet of robotic things, IoRT, Raspberry Pi, REST API</p>		
<p>Miscellaneous (Confidential information)</p> <p>No confidential information</p>		

Sisältö

Lyhenteet.....	6
1 Mobiilirobotti osana robottien internetiä	8
2 Työn toimeksiantajat.....	9
2.1 Transval Goup	9
2.2 Robots on Road –hanke	10
3 Mobiilirobotiikka.....	11
3.1 Mobiilirobotiikka sisälogistiikassa	13
3.2 Mobiilirobotin rakenne	14
3.3 Mobiilirobotin anturointi	15
4 Robotti osana internetiä.....	17
4.1 Esineiden internet, IoT	17
4.2 Pilvirobotiikka	18
4.3 Robottien internet, IoRT	19
5 Sisäpaikannusjärjestelmä, IPS.....	20
5.1 RF-toimisia sisäpaikannusjärjestelmiä	21
5.2 Ultraäänitoimisia sisäpaikannusjärjestelmiä	22
5.3 Infrapunatoiminen sisäpaikannusjärjestelmä.....	24
6 MiR100 mobiilirobotin sarjaliitännän testaus.....	24
6.1 Robotin ja PC:n välinen sarjaliikenne.....	25
6.2 Robotin ja Raspberry Pi korttitietokoneen välinen sarjaliikenne	26
7 IoRT-järjestelmän ja Raspberry Pi:n välinen tiedonsiirto	28
7.1 Raspberry Pi:n verkkoasetukset.....	28
7.2 Raspberry Pi:n web-palvelin	29
8 Mobiilirobotin ja Raspberry Pi:n välinen tiedonsiirto.....	30
8.1 MiR100:n verkkoasetukset	30
8.2 HTTP-viestintä Python Requests komentosarjojen avulla	31
9 IoRT-järjestelmän ja mobiilirobotin välinen tiedonsiirto	32
9.1 HTTP-viestintä cURL komentoriviohjelman avulla.....	32
9.2 IoRT-järjestelmän testaus	34
10 Johtopäätökset	35
11 Pohdinta	36
Lähteet	40
Liitteet	48
Liite 1: Komennot MiR100 sarjaliitännän kautta.....	48
Liite 2: Alustavat Requests-testit.....	49
Liite 3: IoRT-järjestelmän testaus	50

Kuviot

Kuvio 1. Transval Group Oy:n logot	10
Kuvio 2. JAMK Robotics Centerin logo	11
Kuvio 3. Mobiilirobotiikkaa maalla (UGV), ilmassa (UAV), vedessä (USV), sukelluksissa (UUV) ja avaruudessa	12
Kuvio 4. Mobiilirobotiikkaa kunnossapidossa, vartioinnissa, kotitaloustöissä ja terveydenhuollossa	12
Kuvio 5. Amazon Kiva mobiilirobotteja kuljettamassa hyllykköjä	13
Kuvio 6. Differentiaaliset mobiilirobotit MiR100 (vas.) ja GoPal 400 (kesk.) siirtävät tavaraa kuljetusalustan (oik.) tavoin	14
Kuvio 7. Suuntaamaton mobiilirobotti Agilox IGV (vas.) muistuttaa kuljetusmekanismiltaan lavansiirtovaunua (oik.)	15
Kuvio 8. KUKA omniMove liikkuu suuntaamattomasti mecanum-renkaidensa avulla	15
Kuvio 9. Havainne kuva perinteisellä matkamittauksella saadun kartan (a) ja todellisen tilan (b) välisestä erosta	16
Kuvio 10. IoT-järjestelmän rakenteen havainne kuva	18
Kuvio 11. Fetch Robotics AMR-järjestelmän FetchCore-pilvialusta	19
Kuvio 12. IoRT-järjestelmän yhteneväisyydet ja erot verrattuna IoT:iin ja pilvirobotiikkaan ..	20
Kuvio 13. Quuppa IPS-järjestelmän havainnekuva	21
Kuvio 14. UWB signaalin taajuusalue (vas.) ja pulssin pituus (kesk. ja oik.) verrattuna perinteiseen RF-signaaliin	22
Kuvio 15. Active Bat järjestelmän rakenne	23
Kuvio 16. MiR100:n sarjaporttiin liitetty Raspberry Pi	25
Kuvio 17. PuTTYn asetukset nollamodeemia testastattaessa	26
Kuvio 18. TTL232-R-RPi-kaapeli kytkentään käytetyt RasPi:n liittimet (ja johtimet) olivat: 6 (musta), 8 (keltainen) ja 10 (oranssi).	27
Kuvio 19. Raspberry Pi:n ja PC:n välisen sarjaliikenteen testaukseen käytetty komentosarja ..	27
Kuvio 20. Raspberry Pi:n ja MiR100:n välisen sarjaliikenteen testaukseen käytetty komentosarja	28
Kuvio 21. Raspberry Pi:n komentokehote konfigurointitiedoston avaamiseksi ja siihen muokatut parametrit	28
Kuvio 22. Raspberry Pi:n komentokehote verkkorajapintatiedoston avaamiseksi ja sen parametointi	29
Kuvio 23. Langattoman verkon wlan0 parametrit staattisilla IP-osoitteilla.	29
Kuvio 24. Listan 1. ja 2. rivi vastasivat reitittimen ja robotin tietoja vastaavasti	30
Kuvio 25. Havainne kuva MiR100 käyttäjäryhmien hierarkiasta	31
Kuvio 26. Robotin tilatietojen muokkaaminen PUT-metodilla. Rivin 4 tunnistetta on lyhennetty kuvan kaventamiseksi	32
Kuvio 27. Robotin tehtävät listan pyytäminen cURL:lla. Tunnistetta on lyhennetty kuvan kaventamiseksi	33
Kuvio 28. Tehtävän lisääminen tehtäväjonoon. Komentorivin cmd.exe merkistö edellytti JSON-dataosiossa muotoilua.	33
Kuvio 29. Robotin tilatiedon muuttaminen tilaan "ready" (id: 3) ja "pause" (id: 4) sekä virhetilan kuittaus	33
Kuvio 30. Kysely robotille tallennetuista karttapisteistä ja kartoista	34
Kuvio 31. Karttapisteen "transval1" sijainnin päivittäminen. Tehtävän kannalta pisteen alkuperäinen nimi tuli säilyttää	34
Kuvio 32. GET /positions kysely ja vastauksena saatu karttapistettä "transval3" vastaava JSON-olio	34
Kuvio 33. MiR sarjaliitännän mahdollistamat käskyt	48

Taulukot

Taulukko 1. MiR100 ja GoPal 400 mobiilirobottien turva-anturointi	17
Taulukko 2. Ilman lämpötilan vaikutus äänen nopeuteen (Knight 2008, 617)	24
Taulukko 3. IoRT-järjestelmän testisijainnit.....	35

Lyhenteet

AoA	Angle of Arrival, tulokulma (Bluetooth 5.1 tekniikka)
API	Application Programming Interface, ohjelmointirajapinta
BLE	Bluetooth Low Energy, Bluetooth tekniikka
FTDI	Future Technology Devices International, elektroniikka-alan yritys
GPS	Global Positioning System, maailmanlaajuinen satelliittipaikannusjärjestelmä
GUID	Globally Unique identifier, tietojärjestelmissä käytetty 128-bittinen tunniste
HTTP	Hypertext Transport/Transfer Protocol, web-palvelimilla käytetty tiedonsiirtoprotokolla
IoRT	Internet of Robotic Things, <i>"robottien internet"</i>
IoT	Internet of Things, esineiden internet
IPS	indoor positionin system, sisäpaikannusjärjestelmä
JavaScript	komentosarjakieli eli ohjelman suorituksen aikana käännettävä ohjelmointikieli
JSON	JavaScript Object Notation, JavaScript-riippumaton tiedostomuoto
LE	Low Energy, matalaenerginen (Bluetooth 4.0 tekniikka)
LIDAR	valotutka
NPM	Node Package Manager, JavaScript:lle tarkoitettu pakettinhallintajärjestelmä
PC	personal computer, tietokone

PHP	ohjelmointikieli web-palvelinten ohjelmointiin
REST	Representational State Transfer, HTTP-pohjainen ohjelmointirajapinta-arkkitehtuuri
RF	radio frequency, radiotaajuus
RSS	Received Signal Strenght, signaalin vahvuus
RTLS	Real-Time Location System, reaaliaikainen paikannusjärjestelmä
SLAM	Simultaneous Localization and Mapping, SLAM-tekniikka
SSH	Secure Shell, tietoliikenneprotokolla
TDOA	Time Difference of Arrival, saapumisaikaero
TOA	Time of Arrival, (signaalin) saapumisaika
ToF	Time of Flight, (signaalin) lentoaika
TTL232-R-RPi	FTDI:n tuote, USB-UART kaapeli (Raspberry Pi:lle)
UART	Universal Asynchronous Receiver-Transmitter, sarjaliikennepiiri
UAV	Unmannad Aerial Vehicle, ilmassa liikkuva mobiilirobotti
UGV	Unmanned Ground Vehicle, maalla liikkuva mobiilirobotti
USB	Universal Serial Bus, sarjaväyläarkkitehtuuri
USV	Unmanned Surface Vehicle, vedenpinnalla liikkuva mobiilirobotti
UUV	Unmanned Underwater Vehicle, sukelluksissa liikkuva mobiilirobotti

1 Mobiilirobotti osana robottien internetiä

Opinnäytetyön aiheena on mobiilirobotin ohjaus osana esineiden internet eli IoT-järjestelmää (Internet of Things). Mobiilirobotti on määritelmänsä mukaan robotti, joka kykenee kulkemaan sijainnista toiseen itsenäisesti. Esineiden internetiä, jossa robotti on osana, kutsutaan IoRT-järjestelmäksi (Internet of Robotic Things). Käsite on niin uusi, ettei sille ole vielä vakiintunut suomenkielistä ilmaisua. Opinnäytetyö asettuu tukemaan IoRT-järjestelmälle aiemmin ehdotetun termin ”robottien internet” vakiinnuttamista.

Työn aiheesta mielenkiintoisen tekee mahdollisuus automatisoida kaupallisen mobiilirobotin ohjaus yksinkertaisen anturitiedon avulla. Eräänä uutena sovelluskohteena opinnäytetyössä on paikkakoordinaattien tuonti ulkoisesta sovelluksesta mobiilirobotille. Tämä eroaisi merkittävästi nykytekniikasta, jolla mobiilirobotti voidaan kutsua sille ennalta ohjelmoituun pisteeseen, esimerkiksi kutsunapin avulla. Opinnäytetyö tähtää tätä vastoin tilanteeseen, jossa vastaavan kutsunapin paikka määrittyisi vasta kutsumishetkellä.

Mahdollisuus kutsua mobiilirobotti ulkoisen anturitiedon perusteella lisäisi mobiilirobottiympäristön automaatioastetta merkittävästi. Näin mobiilirobottia voidaan hyödyntää itsenäisesti työtehtävissä, jotka ovat aiemmin vaatineet ihmisen läsnäoloa. Tämä toisi säästöjä työvoimakustannuksiin. Toisaalta robotin päämäärän määräytyminen kutsuhetkellä lisäisi mobiilirobottisovellusten joustavuutta yhteistyössä ihmisten kanssa. Mahdollistamalla robotin kutsu sinne missä ihmisen suorittama työvaihe sitä vaatii, ihmistyön keskeytykset lyhentyisivät.

Mobiilirobotin kutsuminen anturitiedon perusteella vaatii toimiakseen robotin paikoitustarkkuutta vastaavan paikoitusanturin. Sisälogistiikkaan keskittyvissä sovelluksissa tämä tarkoittaa tarkkaa sisäpaikannusjärjestelmää. Nämä ovat pitkään olleet kehitysvaiheessa. Uusimpien tekniikoiden käyttöönotto sekä uudistuneet säädökset ovat kuitenkin mahdollistaneet entistä käytännöllisempien sisäpaikannusjärjestelmien tulon markkinoille. Opinnäytetyön jatkokehitysmahdollisuuksia ajatellen opinnäytetyön teoriaosuuteen on sisällytetty kartoitus uusimmista sisäpaikannusjärjestelmistä.

Opinnäytetyön tavoitteena oli testata mobiilirobotin tiedonsiirtorajapintoja robotin paikoituksen ohjaamiseksi ulkoisesta järjestelmästä. Työssä pyrittiin luomaan tarvittava tietopohja mobiilirobottisovellukselle, jossa roska-astian täyttöastetta monitoroiva toimilaite lähettää tarvittaessa tyhjennuspyynnön IoRT-järjestelmään. Järjestelmä käskyttää edelleen mobiilirobottia viemään roska-astian tyhjennyspaikalle. Jatkokehitysideana on muokata järjestelmää toimivaksi niin, että roska-astian paikkatiedot voivat vaihdella mielivaltaisesti.

Tiedonsiirto opinnäytetyössä hyödynnettävän IoRT-järjestelmän ja mobiilirobotin välillä tapahtui internetin yli HTTP-viestein JSON-tiedostomuodossa. Tiedonsiirto toteutettiin suoraan IoRT-järjestelmän ja mobiilirobotin välillä, hyödyntäen robotin ohjauspalvelimen REST ohjelmointirajapintaa. Vaihtoehtoisena toteutuksena kokeiltiin tiedonsiirtoa erillisen korttitietokoneen välityksellä. Tähän käytetty Raspberry Pi liitettiin robottiin sen sarjaliitännän kautta sekä vaihtoehtoisesti langattoman paikallisverkon yli hyödyntäen REST API:a. Tiedonsiirrosta REST API:n yli mielenkiintoisen tekee se, että REST on nykyisin käytetyistä ohjelmointirajapintojen arkkitehtuurimalleista yleisin (Application Programming Interface). Sillä toteutetuissa web-sovelluksissa tiedonsiirtoprotokollana käytetään työssä hyödynnettyä HTTP-protokollaa. Raspberry Pi sovelluksista puolestaan mielenkiintoisen tekevät näissä käytettyjen laitteiden edullisuus sekä ilmaiset ohjelmistot.

Opinnäytetyö tehtiin yhteistyössä Transval Group Oy:n kanssa. Transval on suomalainen logistiikka- ja henkilöstöalan konserni, jonka omistaa Posti Group Oyj. Työssä hyödynnettyä IoRT-järjestelmää testattiin yhteistyössä IT-palveluita tuottavan Trackinno Oy:n kanssa. Opinnäytetyö sivuaa Jyväskylän ammattikorkeakoulu Oy:n ”Robots on Road” hanketta. Hankkeen tarkoituksena on perustaa robotiikan osaamiskeskittymä palvelemaan Keski-Suomen PK-yrityksiä.

2 Työn toimeksiantajat

2.1 Transval Goup

Opinnäytetyön toimeksiantajana toimii Transvalin Teollisuuden Palvelukeskus Hyvinkäällä. Palvelukeskus on osa Transval-konsernia, jonka emoyhtiö on Suomen Transval Group Oy (Kuvio 1). Konserni on suomalainen vuonna 1994 liiketoimintansa

aloittanut osakeyhtiö, jonka päätoimiala on tieliikenteen terminaalitoiminta. (Suomen Transval Oy 2019) Vuonna 2019 Suomen Transval Group Oy:n koko osakekanta siirtyi Posti Group Oy:n omistukseen (Posti ostaa sisälogistiikkayhtiö Suomen Transval Group Oy:n 2018).

Nykyään Transval-konsernin liiketoiminta kattaa logistiikkaan keskittyviä teollisuus-, varastointi-, terminaali- ja konsultointipalveluita sekä henkilöstö- ja myymäläpalveluita (Palvelut yrityksille 2020). Hyvinkäällä sijaitseva Teollisuuden Palvelukeskus aloitti toimintansa vuoden 2019 alussa. Keskuksen toiminta keskittyy teollisuuden projektilogistiikkaan ja materiaalivirtojen hallintaan keskittyviin palveluihin. (Transvalin teollisuuden palvelukeskus Hyvinkäällä on Pohjoismaiden monipuolisin 2019.)

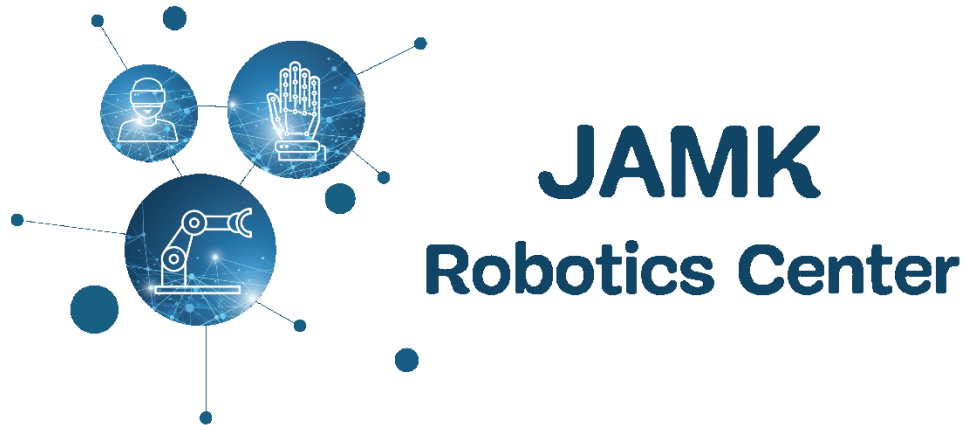


Kuvio 1. Transval Group Oy:n logot (Transval pieni logo; Transval suuri logo)

2.2 Robots on Road –hanke

Opinnäytetyö sivuaa Jyväskylän Ammattikorkeakoulu Oy:n Robots on Road -hanketta. Vuonna 2019 käynnistyneen hankkeen keskiössä on robotiikan osaamiskeskittymän perustaminen osaksi JAMKin liiketoimintaa. Osaamiskeskittymä on nimeltään JAMK Robotics Center (JRC) ja sen tarkoituksena on tarjota keskisuomalaisille PK-yrityksille toimintaympäristö robotiikan testaukseen osana yrityksen kehittyvää liiketoimintaa (Kuvio 2). (JAMK Robotics Center.)

Robotiikan käyttöönottoon liittyy lähes poikkeuksetta suuret alkuinvestoinnit, jotka hankaloittavat pienten ja keskisuurten yritysten mahdollisuuksia robotiikan hyödyntämiseen. JAMK Robotics Center tarjoaa monipuolisen kaluston ja asiantuntijuutta robotiikan kokeiluun yrityksissä (mts). Hanke toteutetaan yhteistyössä paikallisten yritysten kanssa, joihin myös Jyväskylässä toimiva Transval lukeutuu. Osana hanketta toteutetaan pilotointeja yhteistyöyritysten tarpeisiin JRC:n tulevan toimintamallin toimivuuden osoittamiseksi. Opinnäytetyö on yksi näistä piloteista.



Kuvio 2. JAMK Robotics Centerin logo (Minkkinen 2019)

3 Mobiilirobotiikka

Mobiilirobotti on määritelmänsä mukaan robotti, joka kykenee kulkemaan (*en. travel*) itsenäisesti (ISO 8373:2012, 2.13). Käytännössä mobiilirobotti voi olla maalla, ilmassa, veden alla (ja päällä) tai avaruudessa liikkuva (Kuvio 3). Mobiilirobotiikkaa hyödynnetään hyvin erilaisilla toimialoilla, kuten: logistiikassa, autoteollisuudessa, maanpuolustusteollisuudessa, terveydenhuollossa, merentutkimuksessa, avaruudentutkimuksessa ja kotitalouksissa (Kuvio 4). Mobiilirobotiikkaa hyödynnetään esimerkiksi: kunnonvalvonnassa ja kunnossapidossa, turvatehtävissä ja sotilastiedustelussa, koulutuksessa ja tutkimuksessa sekä kotitaloustehtävissä. (Mobile Robotics Market Research Report 2020.) Tämä opinnäytetyö keskittyy sisälogistiikassa tavarankuljettamiseen soveltuvaan mobiilirobotiikkaan.



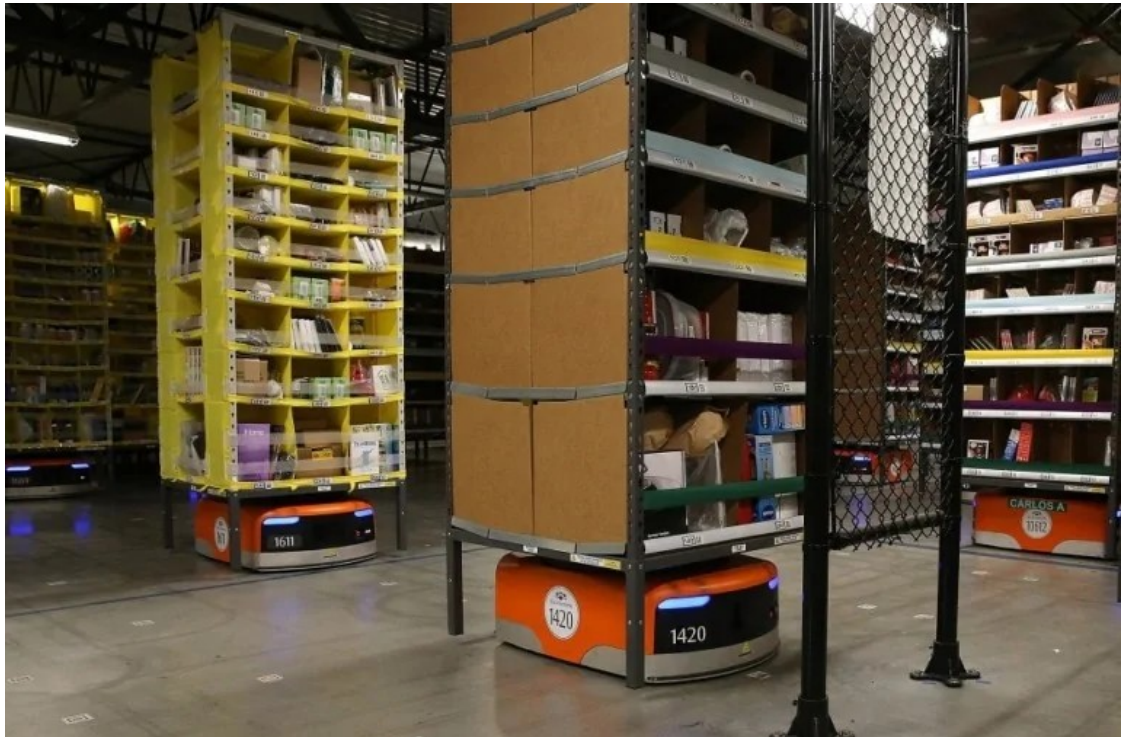
Kuvio 3. Mobiilirobotiikkaa maalla (UGV), ilmassa (UAV), vedessä (USV), sukelluksissa (UUV) ja avaruudessa (Rheinmetall 2019; Altadevices; UTEK-4G 2020; Subsea 7; Sojourner 2011)



Kuvio 4. Mobiilirobotiikkaa kunnossapidossa, vartioinnissa, kotitaloustöissä ja terveydenhuollossa (Falcon; Knightscope; Philips-fc8776; Tuggy)

3.1 Mobiilirobotiikka sisälogistiikassa

Sisälogistiikassa mobiilirobotiikkaa käytetään apuna prosessin eri työvaiheisiin liittyvissä materiaalsiirroissa (Pahlsten 2020). Länsimaissa yhtenä tunnetuimmista esimerkeistä tässä lienevät Amazonin Kiva mobiilirobotit, joita oli käytössä yli 100 000 kappaletta alkuvuodesta 2019 (Kuvio 5) (The story behind Amazon's next generation robot 2019). Loppuvuodesta 2019 robottien lukumäärän uutisoitiin ylittäneen 200 000 kappaleen (O'Brien 2019).



Kuvio 5. Amazon Kiva mobiilirobotteja kuljettamassa hyllykköjä (Overly 2017)

Mobiilirobotiikan yleistymisen sisälogistiikassa on kuitenkin vielä alkuvaiheessa. Vuonna 2018 vajaa 4000 varastoa hyödynsi robotiikkaa (ABI Research 2019). Näistä alle 30 on Amazonin pakkauskeskuksia (What robots do (and don't do) at Amazon fulfillment centers). Viime vuosina mobiilirobottimarkkinat ovat kuitenkin olleet räjähdysmäisessä kasvussa, kun suuret yritykset kuten FedEx, Toyota, Honeywell ja Amazon ovat investoineet mobiilirobotiikkaan. Myös varastotöissä yleistyvän työvoimapulan katsotaan edistävän mobiilirobotiikan käyttöönottoa. Erityisesti verkkokaupan sesonkihuiput korostavat työvoimapulaa jakelukeskuksissa. (Banker 2019; Michel 2019.)

3.2 Mobiilirobotin rakenne

Sisälogistiikassa tavaraa kuljettavat mobiilirobotit muistuttavat kuljetusalustoja tai lavansiirtovaunuja. Mobiilirobotin liikkumismekanismi on tyypillisesti, joko differentiaalikäyttöinen (*en. differential drive*) tai suuntaamaton (*en. omnidirectional*) (Ben-Ari & Mondada 2017, 11). Kummankin mekanismin etuna on robotin kääntymisen mahdollistaminen pienimmällä mahdollisella kääntösäteellä ja ilman erillistä kääntymismekanismia. Suuntaamaton mobiilirobotti kykenee lisäksi liikkumaan lattiatasolla kaikkiin liikesuuntiin kääntymättä. Suurten kuormien kuljetuksiin soveltuviin mobiilirobotteihin on tyypillisesti integroitu nostomekanismi, mikä lisää robotin autonomisuutta materiaalsiirroissa.

Kaupallisina esimerkkeinä differentiaalikäyttöisistä mobiiliroboteista ovat Mobile Industrial Robots MiR100 ja Robotize GoPal 400 (Kuvio 6). Ensin mainitun käsittelykapasiteetti on 100 kg nostamalla ja 300 kg vetämällä (MiR100 User Guide 2017, 23). Jälkimmäisen nostokapasiteetti on 400 kg ja siinä on sisäänrakennettu nostolaite (The GoPal System Operator's Manual 2018, 3).

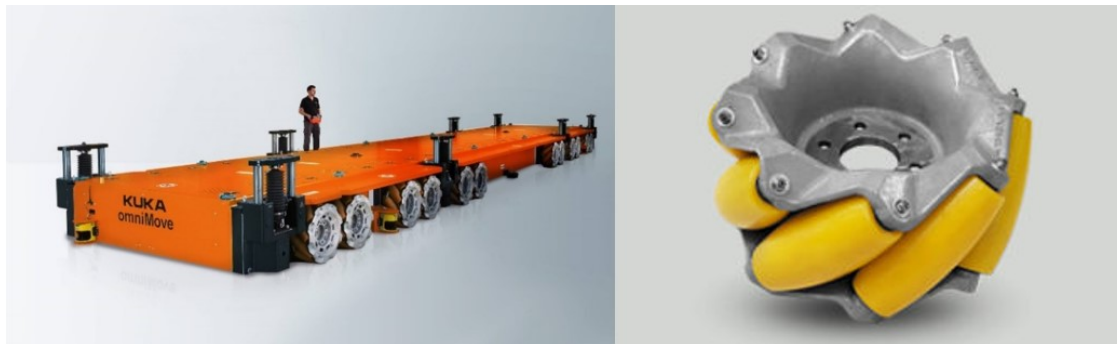


Kuvio 6. Differentiaaliset mobiilirobotit MiR100 (vas.) ja GoPal 400 (kesk.) siirtävät tavaraa kuljetusalustan (oik.) tavoin (MiR100; Robotize; Stanley)

Esimerkkeinä suuntaamattomista mobiiliroboteista ovat Agilox IGV ja KUKA omniMove UTV-2 (Kuvio 7 ja KUKA omniMove liikkuu suuntaamattomasti mecanum-renkaidensa avulla (OmniMove; Mecanum) Kuvio 8). Agilox IGV:n nostokapasiteetti on 1000 kg (Agilox IGV). KUKA omniMove UTV-2:n on suunniteltu hyvin suurten esineiden, kuten lentokoneiden ja tuuliturbiinien siirtoon ja sen suurin mahdollinen nostokapasiteetti on peräti 45 tonnia (KUKA 2017).



Kuvio 7. Suuntaamaton mobiilirobotti Agilox IGV (vas.) muistuttaa kuljetusmekanismiltaan lavansiirtovaunua (oik.) (Banker 2018; Toyota)



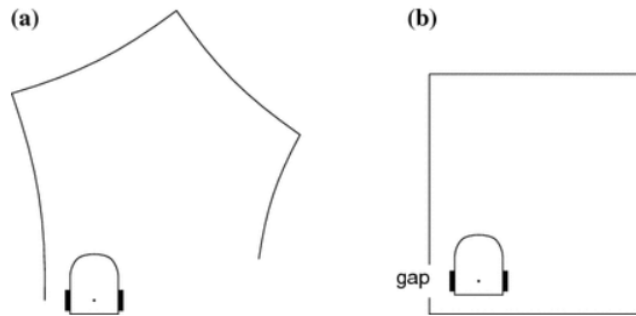
Kuvio 8. KUKA omniMove liikkuu suuntaamattomasti mecanum-renkaidensa avulla (OmniMove; Mecanum)

3.3 Mobiilirobotin anturointi

Mobiilirobotti hyödyntää anturointia navigointiin sekä riittävän laite- ja henkilöturvallisuuden saavuttamiseen. Sisälogistiikkaan soveltuvan mobiilirobotin autonominen paikoitus perustuu SLAM-tekniikkaan (*en. Simultaneous Localization and Mapping*). Tässä robotti luo mittalaitteittensa avulla toimintaympäristöstään kartan, jossa se osaa paikoittaa sijaintinsa vastaavasti anturidatansa avulla (Cadena, Carlone, Carrillo, Latif, Scaramuzza, Neira, Reid & Leonard 2016, 1). Nykyaikaisessa SLAM-tekniikassa yhdistyvät perinteinen matkamittaus, inertiasuunnistus sekä laser ja/tai kuvapohjainen matkamittaus (Scaramuzza & Fraundorfer 2011, 80).

Robotin kulkema matka saadaan mitattua suoraan renkaiden akseleihin liitetyillä pulssiantureilla. Mittaus on paikoitustarkkuuden kannalta virheherkkä. Robotin edetessä sen pulssiantureiden mittausepä-tarkkuus akkumuloituu (Kuvio 9). Lattiatason epätasaisuuksista ja epäpuhtauksista aiheutuva huono kitka robotin ja

lattian välillä saattavat aiheuttaa robotin renkaiden lipsumista, mikä vääristää pulssiantureiden mittaustuloksen oikeellisuutta. (Ben-Ari ym. 2017, 66.)



Kuvio 9. Havainne kuva perinteisellä matkamittauksella saadun kartan (a) ja todellisen tilan (b) välisestä erosta (Ben-Ari ym. 2017, 144)

Inertiasuunnistus tapahtuu robotin IMU-järjestelmän (*en. Inertial Measurement Unit*) avulla. Mobiilirobotin IMU-järjestelmä koostuu kiihtyvyyssantureista ja gyroskoopista. Kiihtyvyyssantureiden avulla mitataan robotin kiihdytykset ja hidastukset. Gyroskoopilla mitataan robotin käännöksiin liittyvät kulmanopeudet. Paikoitus lasketaan kinematiikan perusyhtälöiden avulla. (Ben-Ari ym. 2017, 74–75). Pidemmillä etäisyyksillä myös IMU-järjestelmän virheet akkumuloituvat ja paikoitustarkkuus kärsii.

Laser ja kuvapohjainen matkamittaus ovat tehokas osa SLAM-tekniikkaa (Scaramuzzi ym. 2011, 83–84). Käytännössä kymmenien metrien toimintamatkan, senttimetriluokan mittatarkkuuden, suuren toimintanopeuden ja useiden standardien mukaisten turvaluokitusten vuoksi kaupallisesti merkittävät sisälogistiikan mobiilirobotit käyttävät laserskannereita paikoituksen pääkomponentteina (Turvaskannerit S300 Professional 2020). Kuvapohjaista matkamittausta hyödynnetään usein robotin turva-anturoinnin lisänä, käyttäen mittalaitteina 3D-kameraa (tavallisesti stereo- tai ToF-kamera) (MiR Data sheet 2017; GoPal 400 Data Sheet 2018). Laser ja kuvapohjaisen matkamittauksen heikkoutena ovat optiselle signaalille läpinäkyvät esteet.

Hyvän laite- ja henkilöturvallisuuden saavuttamiseksi mobiilirobottien turva-anturointi on usein redundanttista (Taulukko 1). Esimerkiksi MiR100 mobiilirobotissa optisten antureiden, IMU-järjestelmän ja pulssiantureiden lisänä on eteen ja etuviistoon suunnatut neljä ultraäänianturia (MiR100 User Guide 2017, 18). GoPal

400 mobiilirobotissa vastaava redundanssi on toteutettu koko robotin ulkokehän kattavalla puskurilla, johon on liitetty lähestymiskytkimet (The GoPal System Operator's Manual 2018, 8).

Taulukko 1. MiR100 ja GoPal 400 mobiilirobottien turva-anturointi

Turva-anturi	MiR100 (kpl)	GoPal 400 (kpl)
laserskanneri	2	2
3D-kamera	2	3
ultraäänianturi	4	0
lähestymiskytkin (360°)	0	1

4 Robotti osana internetiä

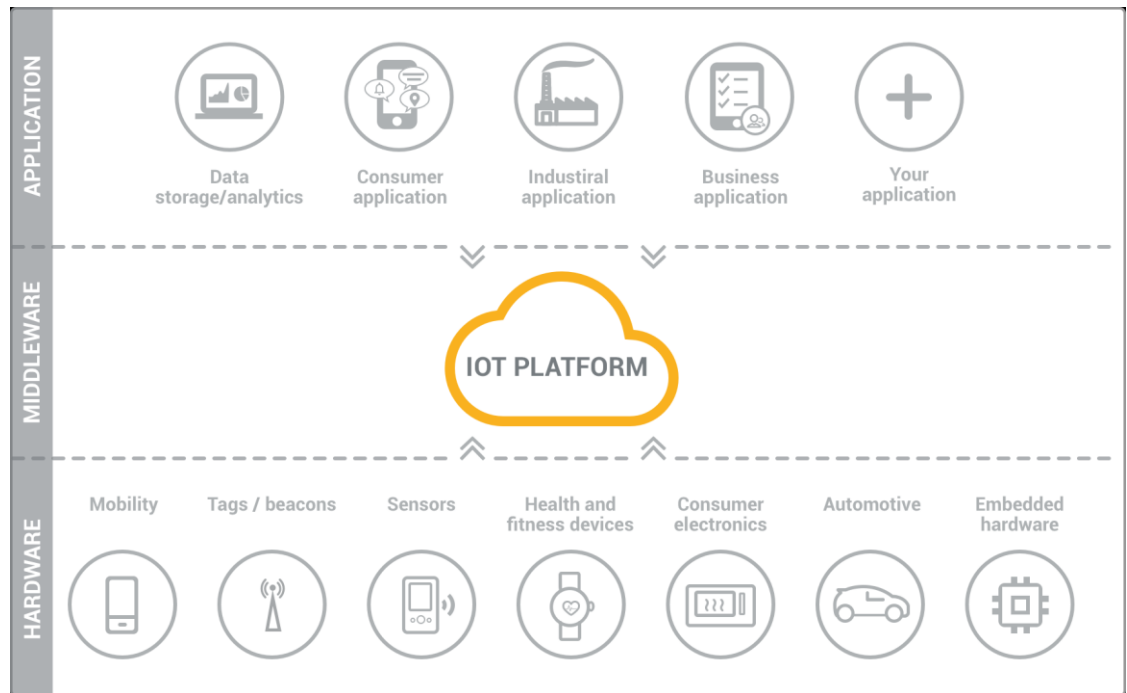
Perinteinen robotin ohjaus tapahtuu paikallisesti robottiohjaimen tai ohjauspalvelimen avulla. Teollisuusroboteilla yksittäiseen ohjaimeen voidaan tavallisesti liittää useampi robotti. Mobiiliroboteilla vastaavasti yksittäiseen ohjauspalvelimeen saatetaan liittää useampi robotti. Kummassakin tapauksessa laitteet muodostavat oman paikallisverkkonsa langallisesti tai langattomasti.

Mahdollisuudet liittää robotti internetiin ovat yleistyneet neljännen teollisen vallankumouksen myötä. Osana teollisuus 4.0:aa myös robotteja on alettu liittää osaksi esineiden internetiä. Samalla myös pilvilaskenta (*en. cloud computing*) on tullut osaksi robotiikkaa. Merkittävänä teollisuuden toimialana robotiikassa näille teollisuus 4.0:n peruskäsitteille on jalostunut omat terminsä: pilvirobotiikka ja robottien internet, IoRT.

4.1 Esineiden internet, IoT

IoT (Internet of Things) eli esineiden internet on järjestelmä, jossa esineet jakavat keskenään tietoa olemassa olevan informaatio- ja kommunikaatiotekniikan avulla. Esineiden internetistä on kehkeytynyt runsaassa 20 vuodessa merkittävä käsite. Vuonna 2017 internetiin liitettyjä esineitä arvioitiin olevan yhteensä noin 8,4 miljardia kappaletta. Arvioitu ei sisällä kannettavia tietokoneita, PC:tä, taulutietokoneita tai matkapuhelimia. (Afanasyev, Mazzara, Chakraborty, Zhuchkov, Maksatbek, Yesildirek, Kassab & Distefano 2019.)

Esineiden internet -verkon keskiössä on IoT-alusta, joka vastaa tiedonsiirrosta verkon anturien ja toimilaitteiden sekä verkkokohtaisten sovellusten välillä. IoT-alusta on kokonaisuutena ikään kuin pilvipalvelimella sijaitseva isäntälaitte, joka hallinnoi verkon orjalaitteina olevia esineitä (Kuvio 10). Samalla alusta toimii ohjelmointirajapintana, joka vastaa kommunikaatiosta käyttäjän ja laitteiden välillä.

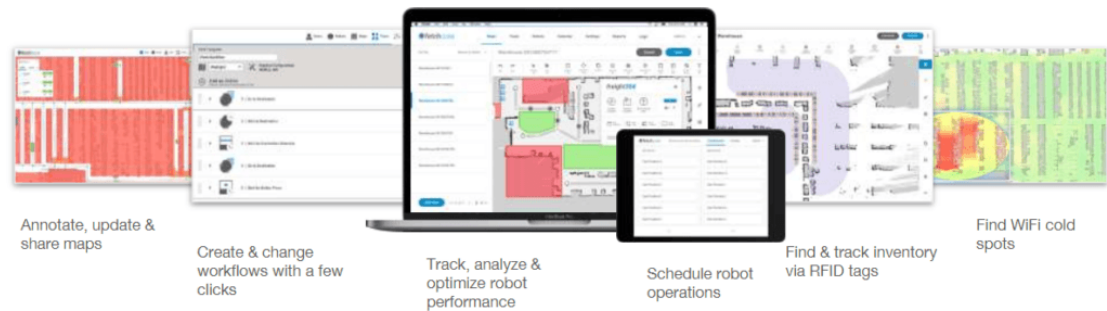


Kuvio 10. IoT-järjestelmän rakenteen havainne kuva (IoT)

4.2 Pilvirobotiikka

Pilvirobotiikassa (*en. cloud robotics*) valtaosa robottiohjaimen laskennasta on ulkoistettu ulkoiselle pilvipalvelimelle. Robotti on yhdistetty palvelimeen langallisella tai langattomalla tiedonsiirtoyhteydellä. (Hu, Tay & Wen 2020, 21.)

Pilviarkkitehtuurin etuina ovat: mahdollisuus ylittää robottiohjaimen prosessorin laskentateho ja muisti, (mobiili)robotin pitkäkestoisempi akunvaraus ja edullisempi rakenne sekä tietokantojen ja kirjastojen tehokkaampi hyödyntäminen (mts. 26). Kaupallisena esimerkkinä pilvipohjaisesta mobiilirobottijärjestelmästä on amerikkalainen Fetch Robotics AMR-järjestelmä (Kuvio 11) (Fetch Robotics 2018).

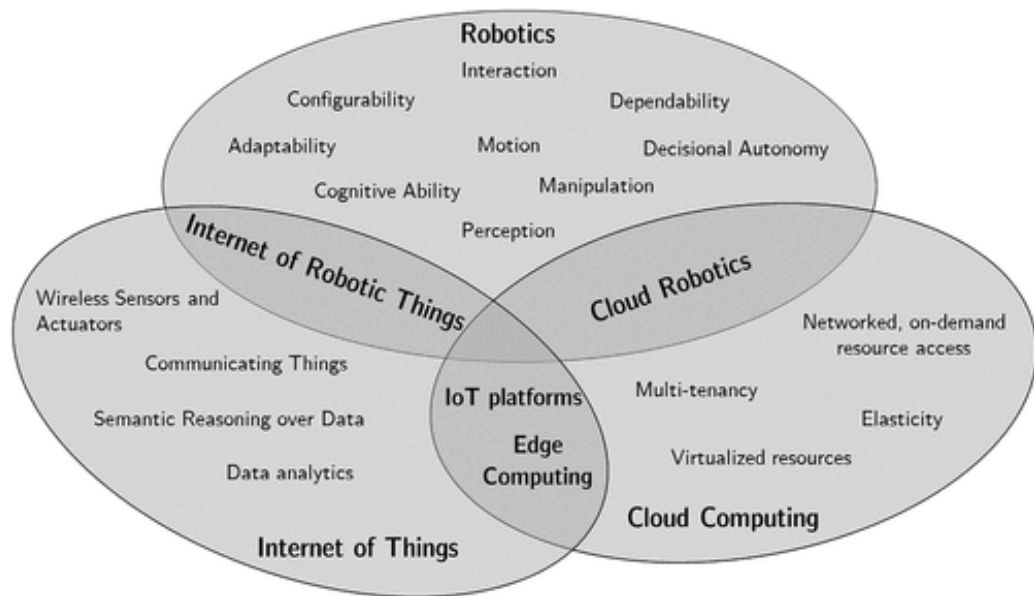


Kuvio 11. Fetch Robotics AMR-järjestelmän FetchCore-pilvialusta (Fetch Cloud Robotics Platform)

4.3 Robottien internet, loRT

Laajennuksena esineiden internetille on luotu uutena käsitteenä loRT (Internet of Robotic Things). Käsite on uusi, eikä sille ole vielä vakiintunut suomenkielistä termiä, mutta Penttilän (2019) mainitsema ”robottien internet” on hyvin linjassa esineiden internetin kanssa. Keskeisenä ajatuksena loRT:ssä on se, että robotit voivat jakaa tietoa keskenään sekä muiden toimintaympäristönsä esineiden kanssa. Näin robotit voivat ylittää oman ohjaimensa määräämät rajoitukset laskentatehossa ja toiminnollisuuksissa. (Ray 2016, 9489–9490.)

loRT muistuttaa pilvirobotiikkaa, mutta sitä on ehdotettu omaksi käsitteekseen esimerkiksi pilvirobotiikan ylittävän laajennettavuutensa vuoksi (mts. 9493). Perinteisestä loT-järjestelmästä loRT-järjestelmä poikkeaa merkittävästi siinä, että roboteilla on kyky muokata (esimerkiksi kuljettaa tai uudelleen järjestää) toimintaympäristönsä fyysisiä rakenteita (Kuvio 12) (Simoens, Dragone & Saffiotti 2018, 5). Mikäli loT-järjestelmän toimilaite varustettaisiin toimimaan vastaavilla vapausasteilla, tultaisiin luoneeksi robotti ja siten loRT-järjestelmä.



Kuvio 12. IoRT-järjestelmän yhteneväisyydet ja erot verrattuna IoT:iin ja pilvirobotiikkaan (Simoens 2018, 2)

5 Sisäpaikannusjärjestelmä, IPS

Sisäpaikannusjärjestelmillä pyritään paikantamaan ihmiset ja asiat rakennusten sisällä. Useimmiten tähän ei voida soveltaa GPS-paikannusta, sillä GPS-satelliittien lähettämä signaali heikkenee merkittävästi rakenteissa (Mäkelä 2008, 73–74). Lisäksi GPS-järjestelmän (Global Positioning System) paikannustarkkuus on verrattain alhainen. Se on sijainnista riippuen parhaimmillaan metrin suuruusluokkaa (Koyuncu & Yang 2010, 126). Useat sisäpaikannuspaikannusjärjestelmät ovat toimintaperiaatteiltaan GPS-paikannusjärjestelmän kaltaisia. Avaruuteen ammuttujen lähettimien sijaan IPS-järjestelmissä (Indoor Positioning System) lähetin kiinnitetään paikannettavaan kohteeseen ja tämän signaalia tarkkailevat vastaanottimet ovat tunnetuilla paikoillaan rakennuksessa. Vastaavasti lähettimet voivat olla myös sisätilan rakenteissa ja vastaanotin paikannettavassa kohteessa.

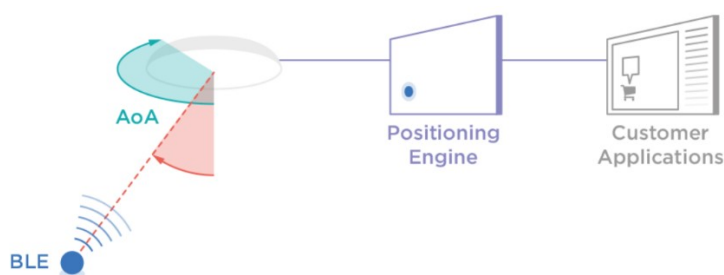
IPS-järjestelmien hyödyntämät signaalityypit vaihtelevat eri kaupallisissa toteutuksissa. Käytettyjä signaalityyppejä ovat esimerkiksi: infrapunasignaali, ultraääni- ja RF-signaalin yhdistelmä, RF-signaali sekä optiset signaalit (konenäkö). (Koyuncu ym. 2010, 121; Mäkelä 2008, 79–81.) Nykyaikaiset kaupalliset ja sisätiloihin soveltuvat RTLS-järjestelmät (real-time location system) hyödyntävät yleisimmin RF-signaalia.

Paikannus perustuu pääasiassa saapumisaikaan (TOA, time of arrival), saapumisaikaeroon (TDOA, time difference of arrival) tai vastaanotetun signaalin vahvuuteen (RSS, Received Signal Strength) (Mäkelä 2008, 37 ja 48). Näistä kahdella ensimmäisellä päästään senttimetriluokan tarkkuuksiin. Viimeksi mainittu tekniikka on herkkä seinien ja muiden esteiden aiheuttamalle signaalin heikkenemälle ja on tarkkuudeltaan parhaimmillaan metrin suuruusluokkaa. (mts. 101.) Tämä opinnäytetyö keskittyy sisäpaikannusjärjestelmiin, joiden paikannustarkkuus on senttimetriluokkaa.

5.1 RF-toimisia sisäpaikannusjärjestelmiä

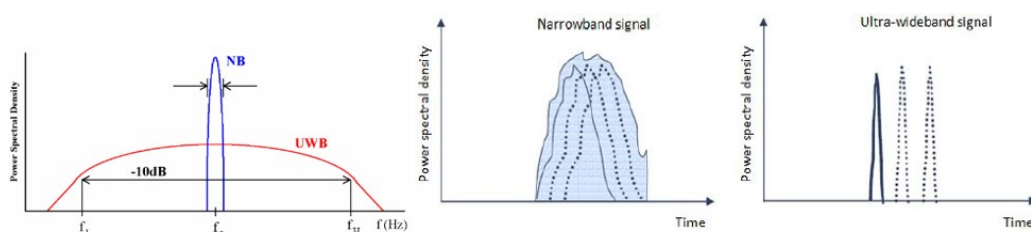
Radiotaajuutta hyödyntävistä kaupallisista RTLS-järjestelmistä esimerkkeinä ovat: Quuppa Intelligent Locating System, Zebra UWB, Sewio UWB ja Eliko KIO RTLS-järjestelmät. Näistä suomalaisen Quuppa Oy:n järjestelmä hyödyntää BLE (Bluetooth Low Energy) ja AoA (Angle of Arrival) tekniikkaa (Quuppa). Amerikkalaisen Zebra Technologiesin, tšekkiläisen Sewio Networks ja virolaisen Eliko Tehnoloogian järjestelmät hyödyntävät UWB-tekniikkaa (Ultra Wide Band) (Zebra 2020; Sewio; Eliko).

Quupan järjestelmässä seurattavaan kohteeseen kiinnitetty tagi lähettää BLE-signaalia, jonka vahvuuden ja tulokulman sisäkattoon kiinnitetty antenni havaitsee. Antenni lähettää anturidatan PC-sovellukselle Ethernet-väylää pitkin (Kuvio 13). Valmistaja lupaa järjestelmälle jopa 10 cm paikoitustarkkuuden. Yksittäisen antennin toiminta-alueeksi luvataan esteisessä ympäristössä 30-50 m, vähäesteisissä teollisuusympäristöissä 70-120 m ja esteettömissä olosuhteissa jopa 300 m. Järjestelmän etuina ovat sen liitettävyyden muihin Bluetooth-laitteisiin, kuten älypuhelimien sekä avoin ohjelmointirajapinta eli API (application programming interface). (Quuppa.)



Kuvio 13. Quuppa IPS-järjestelmän havainnekuva (Quuppa)

Zebran, Sewion ja Elikon IPS-järjestelmät ovat rakenteeltaan Quupan järjestelmän kaltaisia, mutta toimivat UWB-signaalin avulla. UWB-tekniikka poikkeaa perinteisistä RF-tekniikoista. Perinteiset RF-tekniikat hyödyntävät kapealla taajuusalueella lähetettäviä yhtäjaksoisia signaaleja; UWB-signaali on laajalla taajuusalueella hyvin lyhyinä pulsseina lähetetty viesti (Kuvio 14). Tekniikan avulla saavutetaan hyvä TOA tai TDOA mittaustarkkuus ja esteiden läpäisevyys. (Mäkelä 2008, 86–89.) UWB-tekniikalla on mahdollisuus päästä 10 cm paikannustarkkuuteen (Quuppa; Koyuncu ym. 2010, 126; Mäkelä 2008, 101).



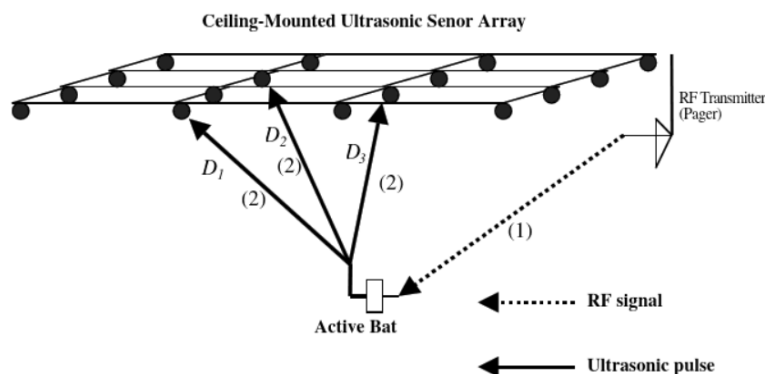
Kuvio 14. UWB signaalin taajuusalue (vas.) ja pulssin pituus (kesk. ja oik.) verrattuna perinteiseen RF-signaaliin (Reed, Buehrer & Ha 2008; Mäkelä 2008, 88)

Zebra lupaa järjestelmälleen esteettömällä signaalilla alle 30 cm paikannustarkkuuden ja korkeintaan 200 m toiminta-alueen. Järjestelmä tukee myös muiden valmistajien standardien mukaisia UWB-tageja. (Zebra UWB Technology 2018.) Sewio lupaa järjestelmälleen 30 cm paikannustarkkuutta ja hyvää skaalattavuutta. Esimerkiksi Volkswagenille tekemässään case-tutkimuksessa järjestelmän toiminta-alueena oli jopa 10 000 m². (Sewio) Eliko lupaa oman järjestelmänsä paikannustarkkuudeksi jopa 5-30 cm (Eliko).

5.2 Ultraäänitoimisia sisäpaikannusjärjestelmiä

Ultraääntä hyödyntävistä järjestelmistä esimerkkeinä ovat: AT&T Cambridgen Active Bat, MIT Laboratoriesin Cricket ja Sonitor. Näistä Active Bat järjestelmässä paikannettavaan kohteeseen kiinnitetään ultraäänilähetin. Vastaanottimien matriisi on sijoitettu toiminta-alueen kattoon ja sen anturien sijainnit tunnetaan tarkasti (Kuvio 15). Lisäksi järjestelmässä on tukiasema, joka lähettää RF-tahdistussignaalin ultraäänilähteelle ja vastaanottimille. Ultraäänilähetin lähettää ultraäänisignaalin vastaanottaessaan RF-signaalin. Vastaanottimet laskevat ultraääni- ja RF-signaalin

välisen saapumisaikojen erotuksen. (Mäkelä 2008, 79-81.) Tukiasema laskee kohteen sijainnin kolmiolaskennalla (Cook & Das 2004, 207).



Kuvio 15. Active Bat järjestelmän rakenne (Cook & Das 2004, 207)

Cricket järjestelmässä paikannettavaan kohteeseen kiinnitetään vastaanotin ja lähettimet ovat kiinteillä sijainneilla, joko katossa tai seinässä. Lähettimet lähettävät kukin vuorollaan ultraääniä ja RF-signaalin. RF-signaali sisältää tiedon lähettimen tunnuksesta ja koordinaateista. Vastaanotin laskee etäisyytensä lähettimestä vastaanotetun ultraääni- ja RF-signaalin saapumisaikojen erotuksesta sekä sijaintinsa kolmiolaskennalla. (Mäkelä 2008, 80.)

Sonitor tarjoaa kaupallisia ultraäänipohjaisia sisäpaikannusjärjestelmiä useilla eri arkkitehtuureilla. Cricket järjestelmän kaltaisten ratkaisujen lisäksi Sonitor tarjoaa järjestelmää, jossa vastaanottimena toimii älypuhelin, tai -kello, tablettitietokone tai PC. (Sonitor 2020.)

Active Bat järjestelmän paikannustarkkuudeksi luvataan 5-10 cm 95 %:n luottamusväillä (Cook ym. 2004, 207). Cricket järjestelmän paikannustarkkuudeksi luvataan jopa 2 cm (Koyuncu ym. 2010, 126). Sonitor ei ilmoita järjestelmilleen täsmällistä paikannustarkkuutta, mutta kyse on ilmeisesti alle 30 cm paikoitustarkkuudesta (Sonitor with ground breaking technology launch).

Ultraäänipohjaisen järjestelmän toimintamatka on alle 10 m. Lyhyehkön kantaman lisäksi ultraäänitekniikka on herkkä ulkoisten äänilähteiden ja rakenteista aiheutuvien heijastusten aiheuttamille häiriösignaaleille. (Mäkelä 2008, 102.) Ultraäänipohjaisen järjestelmän paikannustarkkuutta heikentävää lisäksi ympäristön lämpötilan vaikutus äänennopeuteen (Taulukko 2) (mts. 27). Ultraäänisignaalin etuna on toisaalta se,

ettei se läpäise seinärakenteita. Tämä parantaa paikannustarkkuutta eri huoneiden ja kerrosten välillä. (Sonitor 2020.)

Taulukko 2. Ilman lämpötilan vaikutus äänen nopeuteen (Knight 2008, 617)

Äänen nopeus (m/s)	Ilman lämpötila (°C)
331	0
342	20

5.3 Infrapunatoiminen sisäpaikannusjärjestelmä

Esimerkkinä infrapunasäteilyä hyödyntävästä sisäpaikannusjärjestelmästä on AT&T Cambridgen Active Badges. Järjestelmä on suunniteltu henkilöpaikannukseen.

Paikannettava henkilö pukee infrapunasignaalin lähettävän lähettimen.

Vastaanottimet ovat sijoitettu rakennukseen kiinteille sijainneille. Vastaanotetut signaalit kootaan palvelimelle sijaintitiedoiksi. (Koyuncu ym. 2010, 122.)

Näytteenotto taajuus on valittavissa 10 sekunnin ja 6 minuutin välillä (Cook ym. 2004, 205).

Järjestelmän paikannustarkkuudeksi luvataan 7 cm ja toimintamatkaksi 5 m (Koyuncu ym. 2010, 126). Toisaalta järjestelmän kirjoitetaan olevan kykenemätön kolmiomittaukseen ja sen paikannustarkkuuden kuvaillaan olevan huomattavasti tätä epätarkempi (Cook ym. 2004, 205–206; Mäkelä 2008, 101). Infrapunasäteilyn heikkoutena on sen kantama, joka on suurimmillaan muutama metriä. Lisäksi infrapunasäteily ei läpäise seiniä tai muita esteitä, joten reitti lähettimen ja vastaanottimen välillä tulee olla esteetön. Myös suora auringonvalo ja korkea ympäristön lämpötila aiheuttavat häiriöitä infrapunasignaaliin. (Mäkelä 2008, 26.)

6 MiR100 mobiilirobotin sarjaliitännän testaus

Käytännön testauksissa perehdyttiin mobiilirobotti MiR100:n

tiedonsiirtorajapintoihin. Tavoitteena oli pystyä ohjaamaan robotti haluttuun

sijaintiin ulkoisesta järjestelmästä lähetetyn tiedon avulla. Ensimmäisenä

tiedonsarjarajapintana tutkittiin robotin sisäänrakennettua sarjaliitännää, jonka

kautta robotille voidaan antaa tiettyjä yksinkertaisia komentoja. Sarjaliitännään on

mahdollista kytkeytyä robotin kannessa olevan USB-portin kautta. Mikäli portti on

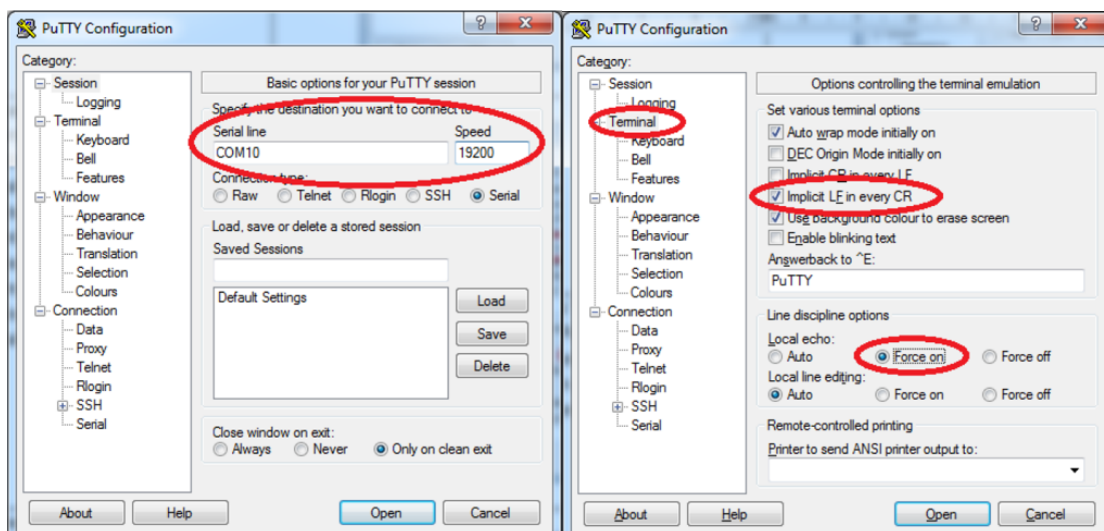
varattu lisävarusteena olevalle 3D-yläkameralle, tulee vaihtoehtoisesti hyödyntää robotin oikeassa etukulmassa olevaa USB-porttia. Sarjaliikenne vaatii toimiakseen tarkoitukseen rakennetun kaapelin. Lisäksi USB-porttiaan käyttävä laite tarvitsee aina käyttötarkoitukseen ohjelmoidun laiteajurin. MiR100 tukee oletuksena Future Technology Devices Internationalin valmistaman sirun sisältäviä sarjaliitântäkaapeleita. (MiR Serial Interface 2017, 3.)

6.1 Robotin ja PC:n välinen sarjaliikenne

Mobile Industrial Robots MiR100:n sarjaliitântää kokeiltiin alustavasti Future Technology Devices Internationalin (FTDI) valmistamalla nollamodeemikaapelilla. Kaapeli kytkettiin kannettavan tietokoneen USB-portista mobiilirobotin etukulmassa vapaana olevaan USB-porttiin (Kuvio 16). Kaapeli toimi lähes täysin kytke ja käytä -periaatteella: Windows 10 käyttöjärjestelmän vaatimat ajurit asentuivat kaapelissa olevan sirun ansiosta automaattisesti. Robotti tunnisti kaapelin ja asensi tarvittavat ajurit, kun sen käyttöliittymän valikossa System/Settings/Serial olevaa hakunäppäintä oli painettu. Mobiilirobotin uudelleen käynnistytksen jälkeen yhteys robottiin avattiin SSH-asiakasohjelma PuTTY:n avulla (Kuvio 17).



Kuvio 16. MiR100:n sarjaporttiin liitetty Raspberry Pi

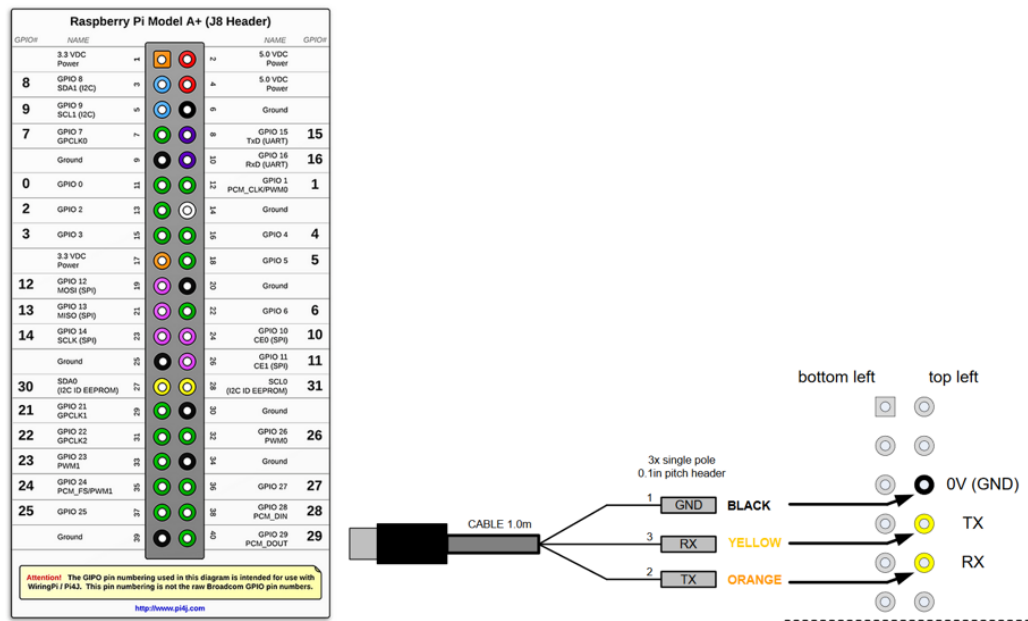


Kuvio 17. PuTTYn asetukset nollamodeemia testattaessa (MiR Serial Interface 2017, 14-15)

Toimintaa testatessa huomattiin sarjaliitännän heikkoudeksi se, ettei robotilla lauennutta virhetilaa (järjestelmätilan ID: 12) voida kuitata sarjaliitännän kautta. Virhetila laukesi yritettäessä ajaa robotti VL-markkerille tai lataustelakalle käskyllä ”!GO:<piste>”, jossa <piste> oli robotin aktiiviseen karttaan tallennetun VL-markkerin tai lataustelakan karttapisteen nimi. Kysyttäessä robotille tallennettujen pisteiden listaa käskyllä ”?L” nämä pisteet näkyivät järjestelmässä kahdesti.

6.2 Robotin ja Raspberry Pi korttitietokoneen välinen sarjaliikenne

MiR100:n sarjaliitännän testausta jatkettiin Raspberry Pi 3 A+ :n ja FTDI TTL232-R-RPi kaapelin avulla. Rasperryn Pi (RasPi) otettiin käyttöön asentamalla MicroSD-kortilla valmiina ostettu Raspbian käyttöjärjestelmä. Samalla käyttäjän ”pi” oletussalasanan ”raspberrry” tilalle vaihdettiin ”V1&nen”. Laite liitettiin JAMK:n langattomaan vierailijaverkkoon. Laitteen ohjelmoimiseksi laitteeseen luotiin etätyöpöytäyhteys VNC-protokollaa hyödyntäen. Tämä onnistui helposti ohjeistuksen perusteella (VNC (Virtual Network Computing) 2020). Lopuksi TTL232-R-RPi-kaapeli kytkettiin Raspberry Pi:n liitinkartan (*en. pinout*) ja kaapelin datalehden mukaisesti (Kuvio 18).



Kuvio 18. TTL232-R-RPi-kaapeli kytkentään käytetyt RasPi:n liittimet (ja johtimet) olivat: 6 (musta), 8 (keltainen) ja 10 (oranssi) (Pin Numbering 2019; TTL-232R-RPi Debug Cable for Raspberry Pi Datasheet 2012).

Ennen robottiin kytkemistä, kaapelin toimintaa testattiin PC:n USB-portin ja PuTTY:n avulla. RasPi:lle ohjelmoitiin yksinkertainen Python-komentosarja, joka ajettiin PC:ltä etätyöpöytäyhteyden avulla. Komentosarjassa hyödynnettiin Raspbian käyttöjärjestelmässä valmiina ollutta pySerial-kirjastoa. Hyödynnetyn Serial() luokan oletuksena käyttämä väylän baudinopeus oli 9600 bps, databittejä 8, pariteetti "none", pysäytysbitti 1 ja aikakatkaisua ei ollut (PySerial 3.0 documentation 2015). Komentosarjassa käytetyn UART-sarjaliikennepiirin nimi "ttyS0" vastasi käytetyn Raspberry Pi mallin sarjaliitännäporttia (Kuvio 19) (Connection to a microcontroller or other peripheral 2019).

```

1 import serial
2 ser = serial.Serial('/dev/ttyS0') # open serial port
3 print(ser.name)                  # check which port was really used
4 ser.write(b'!C')                 # write a string
5 ser.close()                      # close port

```

Kuvio 19. Raspberry Pi:n ja PC:n välisen sarjaliikenteen testaukseen käytetty komentosarja

Komentosarjaa testattaessa huomattiin, että käytetyt RasPi:n liittimet eivät olleet oletuksena osoitettu sarjaliikenteelle (The Raspberry Pi UARTs 2020). Uudelleen konfiguroinnin ja uudelleen käynnistuksen jälkeen sarjaliikenne lähetti PC:n USB-porttiin komentosarjan write() metodin kirjoittaman viestin "!C".

Seuraavaksi sarjaliikennekommunikointia testattiin MiR100:n ja Raspberry Pi:n välillä. Robotille uusi sarjaliikennekaapeli tunnistettiin edellisen tavoin käyttöliittymän etsintäpainikkeella. Tämän jälkeen robotti uudelleen käynnistettiin. Python komentosarja vaati toimiakseen robotilla lisäämään ”\r” (Kuvio 20) (MiR Serial Interface 2017, 6). Testattua komentosarjaa muokattiin lukemaan robotin lähettämät vastaukset io modulin readline() metodia hyödyntäen (Io — Core tools for working with streams 2020).

```
1 import serial
2 import io
3 ser = serial.Serial('/dev/ttyS0', 19600, timeout=1) # open serial port
4 sio = io.TextIOWrapper(io.BufferedRWPair(ser, ser)) #An in-memory stream for text I/O
5 print(ser.name) # check which port was really used
6 ser.write(b'!C\r') # write a string
7 print(sio.readline()) # response (OK: Continue called)
8 ser.close() # close port
```

Kuvio 20. Raspberry Pi:n ja MiR100:n välisen sarjaliikenteen testaukseen käytetty komentosarja

7 IoRT-järjestelmän ja Raspberry Pi:n välinen tiedonsiirto

7.1 Raspberry Pi:n verkkoasetukset

Yhteys Trackinnon IoRT-järjestelmään vaati RasPille julkisen IP-osoitteen. Tämä toteutettiin yhdistämällä RasPi JAMK:n Gadget-verkkoon. JAMK-gadget on JAMK:n verkkolaitteiden oma verkko. Sen SSID on piilotettu.

Raspberry Pi:n yhdistäminen piilotettuun verkkoon ei onnistunut suoraan ”sudo raspi-config” komennon avulla. Ratkaisuna oli muokata langattomien verkkojen tallentamiseen käytettyä wpa_supplicant.conf konfigurointitiedostoa suoraan tekstieditorilla (Kuvio 21). Aiemmin tallennetun JAMK-guest verkon tiedot korvattiin JAMK-gadget verkon tiedoilla. Lisäksi verkon määrittelyyn lisättiin rivi ”scan_ssid=1”, joka tarkoittaa piilotetun verkon hakemista.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
--
network={
    ssid="JAMK-gadget"
    psk="xxx-xxx-xxx"
    scan_ssid=1
}
```

Kuvio 21. Raspberry Pi:n komentokehote konfigurointitiedoston avaamiseksi ja siihen muokatut parametrit

Aluksi yhteyttä testattiin DHCP-palvelimen kanssa. Yhteyden saavuttamiseksi interfaces-tiedosto muokattiin sisältämään tähän tarvittavat rivit. Uudelleen käynnistyksen jälkeen yhteys toimi, vaikka Raspbian työkalupalkin ikoni näyttikin yhteydetöntä tilaa. Lopuksi interface-tiedosto muokattiin vastaamaan järjestelmäsuunnittelijalta saatuja staattisia IP-osoitteita (Kuvio 22 ja Kuvio 23). DNS-osoitteilla ei ollut tässä sovelluksessa käytännön merkitystä, joten ne jätettiin oletusasetusten mukaisiksi. RasPi:n uudelleen käynnistyksen jälkeen yhteys toimi staattisella IP-osoitteella.

```
sudo nano /etc/network/interfaces
--
auto lo
iface lo inet loopback

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

Kuvio 22. Raspberry Pi:n komentokehote verkkorajapintatiedoston avaamiseksi ja sen parametointi

```
auto lo
iface lo inet loopback

auto wlan0
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.111
netmask 255.255.255.0
gateway 192.168.1.1
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

Kuvio 23. Langattoman verkon wlan0 parametrit staattisilla IP-osoitteilla.

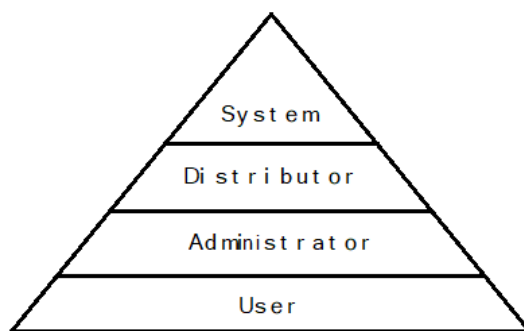
7.2 Raspberry Pi:n web-palvelin

IoT-järjestelmän ja Raspberry Pi:n välinen HTTP-viestintä yritettiin toteuttaa RasPi:lla toimivan web-palvelimen avulla. Olemassa olevista vaihtoehdoista web-palvelimeksi valittiin NGINX. Syynä tähän oli suosituinta Apache2 palvelinta pienempi muistin käyttö (Build your own Raspberry Pi NGINX Web Server 2017). Viimeisimpien ohjelmistoversioiden ja siten parhaan tietoturvan takaamiseksi Raspbian käyttöjärjestelmä yritettiin päivittää versioon Buster. Päivitys epäonnistui 96 % kohdalla.

8.2 HTTP-viestintä Python Requests komentosarjojen avulla

Mobiilirobotin REST API:n käskyttämistä tutkittiin Raspberry Pi:n Python komentosarjojen avulla, hyödyntäen Requests kirjastoa. Requests on HTTP-viestintään kehitetty ja helppokäyttöisyyteen pyrkivä Python ohjelmakirjasto. Kirjasto mahdollistaa JSON-tiedostomuotoisen datan käsittelyn HTTP-viestinnän perus metodeilla. Näistä tärkeimmät ovat GET-metodi tietojen lukemiseen, PUT-metodi olemassa olevan tiedon päivittämiseen ja POST-metodi uusien tietojen kirjoittamiseen. Lisäksi käytettävissä on DELETE-metodi tietojen poistamiseen. (Requests: HTTP for Humans™ 2019.)

HTTP-viestintä robotin ohjelmointirajapinnan kanssa vaati kahdesti kryptatun tunnisteen. Tämä oli mahdollista luoda robotin web-käyttöliittymässä käyttäjätunnuskohtaisesti, käyttöliittymän API-dokumentaatio sivulla olleen toiminnon avulla. Käytetty tunniste mahdollisti robotin samaan käyttäjäryhmään ja tätä hierarkiassa alempana olevien käyttäjäryhmien luomien tietojen muokkaamisen (Kuvio 25). Hierarkiassa ylimpänä oleva käyttäjäryhmä, johon MiR100:n käyttäjällä on oikeudet, on Distributor-ryhmä.



Kuvio 25. Havainne kuva MiR100 käyttäjäryhmien hierarkiasta

Robotin tilatiedot saatiin luettua web-palvelimen osoitteesta <http://mir.com/api/v2.0.0/status> GET-metodilla (Kuvio 26). Saman osoitteen tietojen muokkaaminen PUT-metodilla mahdollisti robotin valmiustilan muuttamisen. Muuttujan "position" arvojen muokkaamisen huomattiin siirtävän robotin sijaan robotin ikonia käyttöliittymän kartalla. Sarjaliitännässä ongelmaksi koettu robotin virhetilan nollaus saatiin toteutettua lähettämällä osoitteeseen JSON-viesti {"clear_error": true} PUT-metodilla.


```

1 import requests
2
3 url="http://[redacted]/api/v2.0.0/status" #JAMK-guest
4 headers={'Authorization':'Basic ZGlzdHJp..AxNA==','Content-Type':'application/json'}
5
6 tila={"state_id": 3} #ready
7 r = requests.put(url, headers=headers, json=tila)
8
9 #robotin ikonin paikka kartalla
10 tila={"position": {"x": 6.5, "y": 15.5, "orientation":0.0}}
11 r = requests.put(url, headers=headers, json=tila)
12
13 tila={"state_id": 4} #pause
14 r = requests.put(url, headers=headers, json=tila)
15 print(r.text)

```

Kuvio 26. Robotin tilatietojen muokkaaminen PUT-metodilla. Rivin 4 tunnistetta on lyhennetty kuvan kaventamiseksi.

9 IoRT-järjestelmän ja mobiilirobotin välinen tiedonsiirto

MiR100 mobiilirobotin ohjelmointirajapinnan tutkimista jatkettiin Raspberry Pi korttitietokoneen sijaan PC:llä. Robotti onnistuttiin käskyttämään haluttuun kartan koordinaattipisteeseen HTTP-viestien avulla. Yksinkertaisimpana toteutuksena tähän nimettiin erillinen karttapiste sekä tehtävä, jossa robotin käskettiin siirtyä vastaavan nimiseen karttapisteeseen. Tämän jälkeen kyseisen karttapisteen koordinaatteja muokattiin halutun laisiksi ja ennalta mainittu tehtävä lisättiin robotin tehtäväjonoon. Robotin liikkeellelähtöä pystyttiin hallitsemaan robotin tilatietoja muuttamalla. Lopuksi menetelmän toimivuutta kokeiltiin IoRT-järjestelmän avulla.

9.1 HTTP-viestintä cURL komentoriviohjelman avulla

Mobiilirobotin ohjelmointirajapinnassa toimivia HTTP-viestejä tutkittiin cURL komentoriviohjelman avulla. Ohjelma on Windows 10 käyttöjärjestelmän komentorivillä toimiva ja HTTP-viestintää tukeva ohjelma tiedonsiirtoon. (Curl 2020.) Ohjelmaa hyödynnetään muun muassa REST ohjelmistorajapintojen toimivuuden testaukseen (Test a REST API with curl 2019).

Robotin siirtoon haluttu toiminnallisuus alustettiin, luomalla karttapiste "transval1" ja tehtävä "transval" robotin graafisen käyttöliittymän avulla. Tehtävä sisälsi vain yhden komennon, joka oli "move to: transval1". Järjestelmä nimeää automaattisesti jokaiselle luodulle karttapisteelle, tehtävällä, kartalle, käyttäjälle, ja niin edelleen oman yksilöllisen GUID-tunnisteen. Tunniste yksilöi HTTP-viestit halutulle kohteelle. Transval-nimistä tehtävää vastaava tunniste selvitettiin lähettämällä GET-pyyntö

robotin ohjainpalvelimen web-osoitteeseen

`http://xxx.xxx.xxx.xxx/api/v2.0.0/missions` (Kuvio 27). Pyyntö antaa listauksen kaikista robotille tallennetuista tehtävistä. Transval-tehtävän GUID-tunniste oli "dfd01655-4736-11ea-b26d-94c6911301a9". Tunnisteen avulla tehtävä lisättiin tehtäväjonoon POST-metodilla (Kuvio 28).

```
curl -X GET "http://192.168.1.178/api/v2.0.0/missions" -H "accept: application/json"
-H "Authorization: Basic ZGl..0YzYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
```

Kuvio 27. Robotin tehtävät listan pyytäminen cURL:lla. Tunnistetta on lyhennetty kuvion kaventamiseksi.

```
curl -X POST "http://192.168.1.178/api/v2.0.0/mission_queue" -H "accept: application/json"
-H "Authorization: Basic ZGl..GJhY2E0YzYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
-H "Content-Type: application/json"
-d '{"mission_id\\":\\"dfd01655-4736-11ea-b26d-94c6911301a9\\",
  \"message\\":\\\"\\\", \"parameters\\":[      ], \"priority\\\": 0, \"fleet_schedule_guid\\\": \\\"\\\"}'
```

Kuvio 28. Tehtävän lisääminen tehtäväjonoon. Komentorivin cmd.exe merkistö edellytti JSON-dataosiossa muotoilua \".

Robotin tilatiedon ollessa "ready" (state_id: 3) robotti lähti toteuttamaan tehtävää heti, kun se oli lisätty tehtäväjonoon. Mikäli robotin tilatietona oli "pause" (state_id: 4) jäi robotti odottamaan "ready" tilaa. Graafisen käyttöliittymän puolella tämä tilanne vastasi robotin tilaa "executing". Vikatilassa oleva robotti ("error", state_id: 12) jäi odottamaan vikatilaa nollaamista. Robotin tilatietoja muokattiin /status osoitteen ja PUT-metodin avulla (Kuvio 29). PUT-metodi mahdollisti yksittäisten parametrien muokkaamisen, POST-metodissa vaaditun jokaisen JSON-olion parametrin kirjoittamisen sijaan.

```
curl -X PUT "http://192.168.1.178/api/v2.0.0/status" -H "accept: application/json"
-H "Authorization: Basic ZGl..zYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
-H "Content-Type: application/json" -d '{"state_id\\": 3}'
```

```
curl -X PUT "http://192.168.1.178/api/v2.0.0/status" -H "accept: application/json"
-H "Authorization: Basic ZGl..zYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
-H "Content-Type: application/json" -d '{"state_id\\": 4}'
```

```
curl -X PUT "http://192.168.1.178/api/v2.0.0/status" -H "accept: application/json"
-H "Authorization: Basic ZGl..zYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
-H "Content-Type: application/json" -d '{"clear_error\\": true}'
```

Kuvio 29. Robotin tilatiedon muuttaminen tilaan "ready" (id: 3) ja "pause" (id: 4) sekä virhetilan kuittaus

Karttapisteen "transval1" sijainnin muokkaamiseksi pisteen GUID selvitetiin GET /positions komennolla (Kuvio 30). Lisäksi tarvittiin sijaintia vastaavan kartan tunniste,

joka selvitettiin GET /maps komennolla. Kummatkin komennoista tuottivat listauksen kaikista robotille tallennetuista karttapisteistä ja kartoista vastaavasti.

```
curl -X GET "http://192.168.1.178/api/v2.0.0/positions" -H "accept: application/json"
-H "Authorization: Basic ZGl..E0YzYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"

curl -X GET "http://192.168.1.178/api/v2.0.0/maps" -H "accept: application/json"
-H "Authorization: Basic ZGl..E0YzYxOWUwZDQ2Yzg3ZTY4OA==" -H "Accept-Language: en_US"
```

Kuvio 30. Kysely robotille tallennetuista karttapisteistä ja kartoista

Transval1-karttapisteen GUID oli "e60c1ecc-4737-11ea-b26d-94c6911301a9".

Pisteen sijaintia muutettiin, lähettämällä uudet koordinaatit PUT-metodilla osoitteeseen ../positions/e60c1ecc-4737-11ea-b26d-94c6911301a9 (Kuvio 31). JSON-tiedostomuodon x,y-koordinaatteja ja kulmaa vastaavat muuttujat olivat muuttujatyypiltään liukuluja.

```
curl -X PUT "http://192.168.1.178/api/v2.0.0/positions/e60c1ecc-4737-11ea-b26d-94c6911301a9"
-H "accept: application/json" -H "Authorization: Basic ZGl..GJhY2E0YzYxOWUwZDQ2Yzg3ZTY4OA=="
-H "Accept-Language: en_US" -H "Content-Type: application/json" -d "{\"name\":\"transval1\",
\"pos_x\":91.65, \"pos_y\":21.45, \"orientation\":20.225, \"type_id\": 0,
\"map_id\":\"58508c4f-5aaa-11e9-8cf7-94c6911301a9\", \"parent_id\":null}"
```

Kuvio 31. Karttapisteen "transval1" sijainnin päivittäminen. Tehtävän kannalta pisteen alkuperäinen nimi tuli säilyttää.

9.2 IoT-järjestelmän testaus

MiR100 robotin ohjausta osana IoT-järjestelmää testattiin yhteistyössä Trackinno Oy:n kanssa. Testauksia varten luotiin oma Users-ryhmän käyttäjä, nimeltään "Trackinno". Käyttäjän käyttäjätunnus oli "IoT" ja salasana "transval". Graafisen käyttöliittymän avustuksella näistä luotu kryptattu tunniste oli

"SW9SVDpIYW1ZjAwYzAzYTExZDFmM2lwNjNiMzllY2M5OGViYzVjZmNlNjZlYWZjZTl1MTA2ODJmNTBkOThkYTQ2NWE4".

Käyttäjätunnusta hyödyntäen testeihin luotiin positio nimeltä "transval3", jonka GUID selvitettiin GET /positions käskyllä (Kuvio 32).

```
curl -X GET "http://192.168.1.178/api/v2.0.0/positions" -H "accept: application/json"
-H "Authorization: Basic SW9..zllY2M5OGViYzVjZmNlNjZlYWZjZTl1MTA2ODJmNTBkOThkYTQ2NWE4"
-H "Accept-Language: en_US"

{
  "guid": "5fc4ee0d-48bb-11ea-a05d-94c6911301a9",
  "map": "/v2.0.0/maps/58508c4f-5aaa-11e9-8cf7-94c6911301a9",
  "name": "transval3",
  "type_id": 0,
  "url": "/v2.0.0/positions/5fc4ee0d-48bb-11ea-a05d-94c6911301a9"
}
```

Kuvio 32. GET /positions kysely ja vastauksena saatu karttapistettä "transval3" vastaava JSON-olio

Trackinno Oy:n kanssa tehtävää testausta varten listattiin neljä sijaintia JAMK:n robotiikkalaboratorio Origossa (Taulukko 3). Näistä sijainnit 1-3 olivat helposti mobiilirobotin saavutettavissa. Sijainti 4 oli robotin kieltoalueella eli robotin kartassa robotilta kielletyksi määritellyllä alueella. MiR100:n käskyttäminen kieltoalueelle laukaisee virhetilan, jonka nollaamista sijainnilla 4 testattiin. Sijaintien x,y-koordinaatit edustavat metrejä robotin kartalla. Robotin kulma edustaa asteita.

Taulukko 3. IoRT-järjestelmän testisijainnit

piste	pos_x	pos_y	orientation
1	91.65	21.45	0
2	91.65	21.45	90
3	90	24	0
4	87.85	19.70	-145

Varsinaisessa testissä Trackinno Oy:n edustaja ohjasi mobiilirobotti MiR100:a IoRT-järjestelmästä käsin internetin yli. Robotti siirtyi testisijaintien 1-3 välillä alustavien testien mukaisesti. Myös sijaintiin 4 käskiessä robotin virhetila laukesi, joka saatiin nollattua IoRT-järjestelmästä lähetetyllä komennolla.

10 Johtopäätökset

Opinnäytetyössä liitettiin MiR100 mobiilirobotti osaksi Transval Group Oy:n IoRT-järjestelmää. Liittäminen todettiin onnistuneeksi ohjaamalla mobiilirobottia IoRT-järjestelmästä käsin, järjestelmän rakentaneen Trackinno Oy:n avustuksella. Robotin ohjaus ulkoisesta järjestelmästä todettiin onnistuneeksi myös paikallisesti, käyttäen robotin kanssa samassa lähiverkossa toimivaa PC:ä sekä Raspberry Pi korttitietokonetta.

Opinnäytetyön tavoitteena oli luoda menettely, jolla ulkoinen järjestelmä pystyi kutsumaan mobiilirobotin haluamaansa sijaintiin. Tämä tavoite saavutettiin, MiR100 mobiilirobotin olemassa olevia toimintoja käyttäen. Onnistuneena menettelynä oli luoda robotin web-käyttöliittymässä robotille karttapiste sekä tehtävä, jossa robotti käskettiin siirtymään vastaavan nimiseen karttapisteeseen. Tämän jälkeen ulkoisesta järjestelmästä voitiin muokata kyseisen karttapisteen koordinaatit halutuiksi.

Lisäämällä ulkoisesta järjestelmästä edellä mainittu tehtävä robotin tehtäväjonoon ja edelleen robotin tilatietoja ohjaamalla robotti saatiin kutsuttua haluttuun sijaintiin.

Työssä toimivaksi osoitetun mobiilirobotin ohjaustavan jatkokehitysmahdollisuuksia tutkittiin selvittämällä nykyaikaisten sisäpaikannusjärjestelmien paikoitustarkkuus ja toiminta-ala. Tehdyn selvityksen perusteella tarkimmat kaupalliset IPS-järjestelmät olivat Eliko Tehnologiaan ja Quuppa Oy:n myymät järjestelmät. Näistä Eliko KIO RTLS toimi UWB-tekniikalla ja sille luvattu paikoitustarkkuus oli jopa 5 cm. Eliko ei ilmoittanut järjestelmänsä toiminta-alaa, mutta vastaavalla tekniikalla toimivien kilpailijoiden järjestelmille luvattiin toiminta-alaksi jopa 10 000 m². Kotimainen Quuppa Intelligent Locating System toimi BLE-tekniikalla ja sille luvattiin 10 cm paikoitustarkkuus ja enimmillään 300 m toimintamatka.

11 Pohdinta

Tärkeimpänä osana mobiilirobotin ohjausta saavutettiin menettely kutsua robotti haluttuun paikkakoordinaattiin, jonka sijaintia voitiin muuttaa ulkoisesta järjestelmästä käsin. Perinteisesti mobiilirobotille ohjelmoidaan ennalta kiinteät paikkakoordinaatit, joihin robotti voidaan kutsua (myös erillisellä laitteella, esimerkiksi kutsunapilla). Opinnäytetyössä toteutettu menettely mahdollistaa paikkakoordinaattien muokkaamisen robotin kutsun yhteydessä, mikä lisää huomattavasti mobiilirobotisovellusten joustavuutta muuttuvissa ympäristöissä.

Yksinkertaisimmillaan tällainen järjestelmä voisi olla esimerkiksi älypuhelinsovellus, jolla sijaintitiedot luetaan rakenteeseen kiinnitetystä QR-koodista ja lähetetään IoT-järjestelmälle yhdellä napin painalluksella. Järjestelmä välittäisi mobiilirobotin kutsun tähän kyseiseen sijaintiin. Tällaisen sovelluksen käytettävyys rajoittuisi luonnollisesti ennalta määriteltyihin sijainteihin, mutta sen välittömänä hyötynä olisi edullisuus perinteisiin kutsunappeihin verrattuna. Kaupallisille teollisuustason mobiiliroboteille erillislaitteina myytävät kutsunapit ovat hintaluokaltaan satoja euroja; paperille tulostettu QR-koodi on hintaluokaltaan senttejä. Älypuhelin löytyy lähes jokaisen työntekijän taskusta valmiina.

Edellä mainitun älypuhelinsovelluksen edullisuutta lisää opinnäytetyössä toteutetun mobiilirobotin ohjauksen helppo skaalattavuus: Sijaintien lukumäärän lisääminen ei

lisää mobiilirobotin ohjelmoinnin tarvetta. Järjestelmä toimii samalla periaatteella niin yhdellä kuin usealla sijainnilla.

Autonomisen toiminnan kannalta ulkoiselta järjestelmältä vaaditaan kyky itsenäisesti selvittää omat sijaintitietonsa robotin toiminta-alueella. Osana opinnäytetyötä kartoitettiin nykyaikaiset sisäpaikannusjärjestelmät. Selvityksen tavoitteena oli löytää tekniikka, jolla sijainnin paikkakoordinaatit voitaisiin välittää mobiilirobotille ulkoisen anturitiedon perusteella.

Sisäpaikannusjärjestelmissä hyödynnetty tekniikka on kehittynyt valtavasti aivan viime vuosina. Tämän hetken tilanne on se, että suhteellisen esteettömässä ja häiriöttömässä tilassa olevan kappaleen sijainti voidaan paikoittaa parhaimmillaan noin 10 cm tarkkuudella. Osa IPS-järjestelmien valmistajista ilmoitti tätäkin tarkempia paikoitustarkkuuksia, mutta nämä olivat ristiriidassa muiden vastaavien järjestelmien ja alan kirjallisuuden kanssa. Opinnäytetyössä tapausesimerkkinä olleen roska-astian tyhjennystehtävän tapauksessa, tällainen 10 cm paikannustarkkuus saattaisi olla riittävä.

Käytännössä vaadittaisiin erillinen ulkoinen järjestelmä, jolla näin saavutettu karkea sijainti täsmennettäisiin roska-astian mobiilirobottiin kiinnittämiseksi. Tällaisena saattaisi toimia robottiin kiinnitetty mekaaninen toimilaite. Tämä kykenisi kiinnittymään roska-astiaan, kun sen sijainti ja mobiilirobotin sijainti ovat riittävän lähellä toisiaan.

Toisaalta tällaisena erillisenä hienopaikannusjärjestelmänä voisi toimia anturisignaaliin perustuva robottiin kiinnitetty ohjausjärjestelmä, joka pystyisi havaitsemaan roska-astian lähietäisyydeltä. Tällainen järjestelmä voisi käskyttää robottia signaalin suuntaan sopivan pienin askelluksin, kunnes roska-astiaan pitelevän toimilaitteen kiinnittäminen olisi mahdollista. Ohjausjärjestelmä toimisi ohjelmallisesti täysin samalla periaatteella, kun mitä opinnäytetyössä käytettiin robotin kutsumiseksi haluttuun sijaintiin. Nyt sijainti vain päivittyisi taajemmin etäisyysanturitiedon perusteella.

Käytännössä opinnäytetyössä toteutettu menettely toimii sellaisenaan vain MiR100 mobiilirobotille. Tämä johtuu siitä, että menettely toteutettiin kyseisen robottiin ja

sen REST API:in rakennettuja ominaisuuksia hyödyntäen. Mikäli vastaavat toiminnallisuudet ovat käytettävissä toisen robottivalmistajan järjestelmässä, on vastaava toimintatapa mahdollinen. Yleisesti ottaen web-selainpohjaisella graafisella käyttöliittymällä toimiva mobiilirobotti hyödyntää jonkinlaista ohjelmointirajapintaa käyttäjän käyttöliittymälle antamien tietojen siirtämiseksi käyttöliittymältä robotin ohjainpalvelimelle. Nykyisin käytettyjen ohjelmistorajapintojen arkkitehtuurimalleista REST on eniten käytetty. Kaupallisen mobiilirobotin ohjelmointirajapinta on näin ollen todennäköisimmin REST API. Mikäli ohjelmointirajapinta sallii siihen yhteydenoton ulkoisesta järjestelmästä, voidaan opinnäytetyön menettelyä vastaava toiminnallisuus toteuttaa.

Robotin ohjausta ulkoisesta järjestelmästä voidaan pitää yhtä tietoturvallisena kuin tähän käytettyjen järjestelmän ja tietoverkon tietoturvaa. Robotin ohjaaminen opinnäytetyössä toteutetulla menettelyllä REST API:n avulla vaatii aina käyttäjän tunnistuksen. Tämä on toteutettu MiR100:n käyttöliittymään luodusta käyttäjätunnuksesta ja sen salasanasta muodostetulla kahdesti kryptatulla tunnisteella. Tunnisteen pituus on yli 100 merkkiä, minkä ansiosta sen murtaminen väsytyksen menetelmällä (*en. brute force method*) on hyvin vaikeaa. Tunnisteen tietoturvallisuus edellyttää luonnollisesti sen, ettei hakkeri tunne käytettyjä kryptausmenetelmiä ja käyttäjätunnuksen salasanaa.

Käytännön testaukset jäivät ulkoisten järjestelmien osalta vähäisiksi opinnäytetyön toteutuksessa. Syinä tähän olivat opinnäytetyön tekijän päivätyöstä johtuvat aikataululliset haasteet sekä toimeksiantajan omistajuuden vaihdoksen aiheuttamat kiireet. Oman kirsikkansa kakun päälle toi COVID-19 pandemia, joka uhkasi keskeyttää kaiken robotilla tehtävän käytännön työskentelyn. Syntyneistä viiveistä johtuen opinnäytetyö päätettiin valmistella suunnitellussa aikataulussa, ilman lisää käytännön testauksia.

Pandemian mahdollistaman muuttuneen työjärjestyksen seurauksena kuitenkin onnistuttiin ideoimaan käytännön koejärjestely, jossa opinnäytetyön tuloksia sovelletaan. Tarkoituksena on kokeilla mobiilirobotin ohjausta kattoon kiinnitetyltä konenäkökameralta saatujen paikkakoordinaattien avulla. Tavoitteena on pystyä

käskyttämään robotti haluttuun sijaintiin riittävän tarkasti, ilman erillistä ulkoista hienopaikannusjärjestelmää.

Järjestelmä muistuttaisi toiminnaltaan teollisuusnivelsirobottien kappaleenkäsittelyssä jo pitkään käytettyä konenäköjärjestelmää. Mobiiliroboteilla vastaavaa järjestelmää ei tällä tietoa ole kuitenkaan vielä yleisesti sovellettu. Onnistuessaan menetelmä loisi uusia mahdollisuuksia automatisoida sisälogistiikan materiaalin siirtotehtäviä. Koejärjestely toteutetaan osana ”Robots on Road” hanketta yhteistyössä Transval Group Oy:n kanssa.

Lähteet

ABI Research. 2019. 50,000 Warehouses to Use Robots by 2025 as Barriers to Entry Fall and AI Innovation Accelerates 26.3.2019. Viitattu 16.1.2020.

<https://www.abiresearch.com/press/50000-warehouses-use-robots-2025-barriers-entry-fall-and-ai-innovation-accelerates/>.

Afanasyev, I., Mazzara, M., Chakraborty, S., Zhuchkov, N., Maksatbek, A., Yesildirek, A., Kassab, M. & Distefano, S. 2019. Towards the Internet of Robotic Things: Analysis, Architecture, Components and Challenges. ResearchGate 8.7.2019. Viitattu 28.2.2020.

https://www.researchgate.net/publication/334360894_Towards_the_Internet_of_Robotic_Things_Analysis_Architecture_Components_and_Challenges.

Agilox IGV. N.d. Agilox mobiilirobotin tekniset tiedot. Viitattu 28.2.2020.

<https://www.agilox.net/en/agilox-igv#key-facts-features>.

Altadevices. N.d. Ilmassa kulkeva mobiilirobotti. Viitattu 16.1.2020.

<https://www.altadevices.com/wp-content/uploads/2017/12/Final-Image2.jpg>.

Application Programming Interface. N.d. Artikkel Robotize ohjelmistorobotiikka yrityksen sivulla. Viitattu 14.5.2020. <http://robotize.co/apification/>

Arp. 2017. Arp-komennon dokumentaatio Microsoftin sivustolla 16.10.2017.

Viitattu 16.12.2019. <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/arp>.

Banker, S. 2018. Agilox Robots Rely on Swarm Intelligence. Forbes 19.6.2018. Viitattu 28.2.2020. <https://www.forbes.com/sites/stevebanker/2018/07/19/agilox-robots-rely-on-swarm-intelligence/#149232fc46e4>.

Banker, S. 2019. The Autonomous Mobile Robot Market Is Taking Off Like A Rocket Ship. Forbes 11.3.2019. Viitattu 16.1.2020.

<https://www.forbes.com/sites/stevebanker/2019/03/11/the-autonomous-mobile-robot-market-is-taking-off-like-a-rocket-ship/#5ac9b1341603>.

Ben-Ari, M. & Mondada, F. 2017 Elements of Robotics. Cham: Springer.

Build your own Raspberry Pi NGINX Web Server. 2017. Ohjeartikkeli PyMyLifeUP sivustolla 27.8.2017. Viitattu 9.12.2019. <https://pimylifeup.com/raspberry-pi-nginx/>.

Cadena, C., Carlone L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. Leonard, J. 2016. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. IEEE Transactions on Robotics, 32, 6, 1309–1332.

Connection to a microcontroller or other peripheral. 2019. Artikkelin eLinux wiki sivustolla. Viitattu 23.3.2020.
https://www.elinux.org/RPi_Serial_Connection#Connection_to_a_microcontroller_or_other_peripheral.

Cook, D. & Das, S. 2004. Smart Environments: Technology, Protocols, and Applications. Hoboken: John Wiley & Sons.

Curl. 2020. Komentoriviyökalu cURL:n kotisivusto. Viitattu 25.3.2020.
<https://curl.haxx.se/>.

Eliko. N.d. Eliko Tehnoloogian kotisivusto. Viitattu 25.2.2020. www.eliko.ee.

Falcon. N.d. Kunnossapidossa käytetty mobiilirobotti. Viitattu 16.1.2020.
https://s2.manualzz.com/store/data/040882668_1-74df074f566a8dad4b8f3341f3064fec.png.

Fetch Cloud Robotics Platform. 2020. Pilvialustan kuvaus Storage Solutions –nimisen jälleenmyyjän kotisivustolla. Viitattu 27.2.2020. <https://storage-solutions.com/warehouse-optimization/dynamic-warehouse-solutions/fetch-robotics/fetchcore/>.

Fetch Robotics. 2018. Fetch Robotics kotisivusto. Viitattu 27.2.2020.
<https://fetchrobotics.com/>.

GoPal 400 Data Sheet. 2018. GoPal 400 mobiilirobotin datalehti.

Hu, G., Tay, W. & Wen, Y. 2020. Cloud Robotics: Architecture, Challenges and Applications. IEEE Network, 26, 3, 21–28.

Io — Core tools for working with streams. 2020. Python io-moduulin dokumentaatio. Viitattu 11.11.2019. <https://docs.python.org/3/library/io.html#module-io>.

IoT. N.d. Viitattu 28.2.2020.

https://miro.medium.com/max/2220/1*r4YKY2YckQgRyGEtb-l80w.png.

ISO 8373:2012(en). Robots and robotic devices — Vocabulary. Aihealueet: Robottien ja robotisoitujen laitteiden termistö. Vahvistettu 3/2012. Viitattu 16.1.2020. <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>.

JAMK Robotics Center. Tulevaisuuden robotiikkaa kehittämässä. N.d. Jyväskylän ammattikorkeakoulun toteuttaman projektin kuvaus jamk.fi kotisivuilla. Viitattu 19.3.2020. <https://www.jamk.fi/fi/Tutkimus-ja-kehitys/projektit/jamk-robotics-center/jamk-robotics-center/>.

Knight, R. 2008, Physics for Scientists and Engineers: A Strategic Approach with Modern Physics. 2 p. Pearson.

Knightscope. N.d. Vartiinnissa käytetty mobiilirobotti. Viitattu 16.1.2020.

[https://cdn.vox-cdn.com/thumbor/EB8vOqOjInb81nsieYfvydrAJLM=/0x0:1000x667/1220x813/filters:focal\(420x254:580x414\):format\(webp\)/cdn.vox-cdn.com/uploads/chorus_image/image/65700241/IMG_1635.0.0.jpg](https://cdn.vox-cdn.com/thumbor/EB8vOqOjInb81nsieYfvydrAJLM=/0x0:1000x667/1220x813/filters:focal(420x254:580x414):format(webp)/cdn.vox-cdn.com/uploads/chorus_image/image/65700241/IMG_1635.0.0.jpg).

Koyuncu, H. & Yang, S. 2010. A Survey of Indoor Positioning and Object Locating Systems. International Journal of Computer Science and Network Security, 10 5, 121–128.

KUKA. 2017. KUKA omniMove UTV-2 datalehti.

Mecanum. N.d. Mecanum-renkaan havainnekuva. Viitattu 28.2.2020.

<https://image.made-in-china.com/43f34j00IJZEefMILBbv/Diameter-310mm-Mecanum-Wheel-for-Robot-Platform.jpg>.

Michel, R. 2019. Mobile Robotics Descend on Warehouse Operations. Logistics Management 10.6.2019. Viitattu 16.1.2020.

https://www.logisticsmgmt.com/article/mobile_robotics_descend_on_warehouse_operations.

Minkkinen, T. 2019. RoboCenter logo vaalealle taustalle. Projektiryhmälle jaettu tiedosto. Viitattu 19.3.2020.

<https://jamkstudent.sharepoint.com/sites/RobotsonRoad/Shared%20Documents/General/Ty%C3%B6paketti%205%20-%20Tiedottaminen%20ja%20raportointi/Center%20logot/RoboCenter%20logo%20vaalealle%20taustalle.png>.

MiR100. N.d. MiR100 mobiilirobotti. Viitattu 28.2.2020. <https://mir-development.azureedge.net/cache/4/0/3/d/f/9/403df9d7758d8c18cafe754bbc27d3a85d3050d7.png>.

MiR100 Data sheet. 2017. MiR100 mobiilirobotin datalehti.

MiR Serial Interface. 2017. MiR tekninen dokumentaatio versio 1.7 14.6.2017.

MiR100 User Guide. 2017. MiR100 mobiilirobotin käyttöohje 11/2017 v.1.0.

Mobile Robotics Market Research Report. Forecast 2023. 2020. Market Research Futuren toimittama markkinatutkimus. Viitattu 16.1.2020.

<https://www.marketresearchfuture.com/reports/mobile-robotics-market-1946>.

Mäkelä, P. 2008. Localpositioning systems and indoor navigation. Lisensiaattityö. Tampereen tekninen yliopisto, Sähkötekniikan koulutusohjelma.

O'Brien, M. 2019. As robots slowly take over Amazon's warehouses, are they causing more harm than good? Independent 30.12.2019. Viitattu 18.1.2020.

<https://www.independent.co.uk/news/business/robots-amazon-delivery-artificial-intelligence-technology-a9264036.html>.

OmniMove. N.d. KUKA omniMove mobiilirobotin koon havainnekuva. Viitattu 28.2.2020. <https://www.kuka.com/-/media/kuka->

corporate/images/products/mobility/kuka_omnimove_header.jpg?rev=7098d9f177524c89afa013c45d39114d&w=1900&hash=32F2C9F4B29BF94FC5A74645F3DA0778.

Overly, S. 2017. Why Amazon's use of self-driving technology would be a game changer. The Washington Post 25.4.2017. Viitattu 18.1.2020.

<https://www.washingtonpost.com/news/innovations/wp/2017/04/25/why-amazons-use-of-self-driving-technology-would-be-a-game-changer/>.

Pahlsten, V. 2020. Logistiikan termistöä. Sähköpostiviesti 20.1.2020.

Palvelut yrityksille. 2020. Artikkelit Transval konsernin sivustolla. Viitattu 19.3.2020.
<https://www.transval.fi/palvelut/>.

Penttilä, A. 2019. Robotit tulevat sisälogistiikkaan. Artikkelit Transval konsernin kotisivustolla 8.11.2019. Viitattu 26.2.2020.
<https://www.transval.fi/2019/11/08/robotit-tulevat-sisalogistiikkaan/>.

Philips-fc8776. N.d. Kotitaloustöissä käytetty mobiilirobotti. Viitattu 16.1.2020.
<https://scale.coolshop-cdn.com/product-media.coolshop-cdn.com/AF25Z8/4fe78d0693484731ae7fe0177ed2f11d.jpg/f/philips-fc8776-01-robotic-vacuum-cleaner.jpg>.

Pin Numbering. 2019. Raspberry Pi Model A+ liitinkartta. Viitattu 11.11.2019.
<https://pi4j.com/1.2/pins/model-a-plus.html>.

Posti ostaa sisälogistiikkayhtiö Suomen Transval Group Oy:n. 2018. Tiedote Transval konsernin sivustolla 28.9.2018. Viitattu 19.3.2020.
<https://www.transval.fi/2018/09/28/tiedote-29082018/>.

PySerial 3.0 documentation. 2015. Python pySerial-kirjaston dokumentaatio. Viitattu 11.11.2019. <https://pythonhosted.org/pyserial/shortintro.html#readline>.

Quuppa. N.d. Quuppa Oy:n kotisivusto. Viitattu 25.2.2020. www.quuppa.com.

Ray, P. 2016. Internet of Robotic Things: Concept, Technologies, and Challenges. IEEE Access, 4, 9489-9500.

Reed, J., Buehrer, M. & Ha, D. 2008. Introduction to UWB: Impulse Radio for Radar and Wireless Communication. Diaesitys. Virginia Tech. Viitattu 25.2.2020. <http://sss-mag.com/pdf/uwbcars.pdf>.

Requests: HTTP for Humans™. 2019. Python Requests ohjelmakirjaston dokumentaatio. Viitattu 24.3.2020. <https://requests.readthedocs.io/en/master/>.

Rheinmetall. 2019. Maan pinnalla kulkeva mobiilirobotti. Viitattu 16.1.2020. <https://armadainternational.com/wp-content/uploads/2019/08/Rheinmetall-Mission-Master-Protection-696x465.png>.

Robotize. N.d. GoPal 400 mobiilirobotti. Viitattu 28.2.2020. https://wevolver-project-images.s3.amazonaws.com/0.fwhvwgj1eb02_ANGLE_1_WITH_LIFT.jpg.

Scaramuzza, D. & Fraundorfer, F. 2011. Visual Odometry Part I: The First 30 Years and Fundamentals. IEEE Robotics & Automation Magazine, 8.12.2011, 80–92.

Sewio. N.d. Sewio Networks kotisivusto. Viitattu 25.2.2020. www.sewio.net.

Simoens, P., Dragone, M. & Saffiotti, A. 2018. The Internet of Robotic Things: A review of concept, added value and applications. International Journal of Advanced Robotic Systems, 15, 1, 1–11.

Sojourner. 2011. Avaruudessa kulkeva mobiilirobotti 29.3.2011. Viitattu 16.1.2020. https://upload.wikimedia.org/wikipedia/commons/3/3a/Sojourner_on_Mars_PIA01122.jpg.

Sonitor. 2020. Sonitor yrityksen kotisivusto. Viitattu 24.2.2020. www.sonitor.com.

Sonitor with ground breaking technology launch. N.d. Uutinen Norway Health Tech sivustolla. Viitattu 24.2.2020. <https://www.norwayhealthtech.com/news/sonitor-ground-breaking-technology-launch/>.

Stanley. N.d. Kuljetusalusta Stanley SXWTC-PC508. Viitattu 28.2.2020. <https://www.verkter.fi/kuljetusalusta-stanley-sxwtc-pc508-150-kg.html>.

Subsea 7. N.d. Vedessä sukeltava mobiilirobotti. Viitattu 16.1.2020. https://miro.medium.com/max/1024/1*-qYJ5VtGm2JlxGgjYFlcMg.jpeg.

Suomen Transval Oy. N.d. Yritystiedot Yritys- ja yhteisötietojärjestelmän sivuilla.

Viitattu 19.3.2020.

<https://tietopalvelu.ytj.fi/yritystiedot.aspx?yavain=624528&tarkiste=4B01DC7BBE0A9D1A392BC500181093A33DC21328>.

Test a REST API with curl. 2019. Artikkelin Baeldung sivustolla 4.9.2019. Viitattu

25.3.2020. <https://www.baeldung.com/curl-rest>.

The GoPal System Operator's Manual. 2018. GoPal mobiilirobotin käyttöohje 1.8.2019.

The Raspberry Pi UARTs. 2020. Yhteisön ylläpitämä dokumentti Raspberry Pi yhteisön kotisivustolla. Viitattu 11.11.2019.

<https://www.raspberrypi.org/documentation/configuration/uart.md>.

The story behind Amazon's next generation robot. 2019. Amazon Day One blogikirjoitus 11.3.2019. Viitattu 18.1.2020.

<https://blog.aboutamazon.com/innovation/the-story-behind-amazons-next-generation-robot>.

Toyota. N.d. Lavansiirtovaunu Toyota BT Levio. Viitattu 28.2.2020. <https://toyota-forklifts.fi/globalassets/inriver/resources/toyota-bt-levio-s-series-main.jpg?preset=small>.

Transval pieni logo. N.d. Transval konsernin logo. Viitattu 27.4.2020. https://media-exp1.licdn.com/dms/image/C4E0BAQF940a16KLClw/company-logo_200_200/0?e=2159024400&v=beta&t=ZfX0mFEck5lqX09KzZQjVKKBcQAWuQTVuARE2ygTErM.

Transval suuri logo. 2019. Transval konsernin logo. Viitattu 27.4.2020.

https://www.transval.fi/wp-content/uploads/2019/10/transval_logo.png.

Transvalin teollisuuden palvelukeskus Hyvinkäällä on Pohjoismaiden monipuolisin.

Artikkeli Transval konsernin sivustolla 8.2.2019. Viitattu 19.3.2020.

<https://www.transval.fi/2019/02/08/transvalin-teollisuuden-palvelukeskus-hyvinkaalla-pohjoismaiden-monipuolisin/>.

TTL-232R-RPi Debug Cable for Raspberry Pi Datasheet. 2012. TTL-232R-RPi sarjaliitântäkaapelin datalehti. Viitattu 11.11.2019.

https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_RPi.pdf.

Tuggy. N.d. Terveidenhuollossa käytetty mobiilirobotti. Viitattu 16.1.2020.

<https://media.wired.com/photos/5932842ef682204f73696681/master/pass/tuggy-inline.jpg>.

Turvaskannerit S300 Professional. 2020. Mobiilirobotiikassa käytettyjen turvaskannereiden tekniset tiedot SICK:n kotisivustolla. Viitattu 28.2.2018.

https://www.sick.com/fi/fi/valosaehkoeiset-turvalaitteet/turvalaserskannerit/s300-professional/s30b-2011da/p/p53843?ff_data=JmZmX2lkPXA1Mzg0MyZmZl9tYXN0ZXJJZD1wNTM4NDMmZmZfdGl0bGU9UzMwQj0yMDExREEmZmZfcXVlcnk9JmZmX3Bvcz0xJmZmX29yaWdQb3M9MSZmZl9wYWdlPTEmZmZfcGFnZVNpemU9MjQmZmZfb3JpZ1BhZ2VTaXplPTI0JmZmX3NpbWk9OTQuMA.

UTEK-4G. 2020. Vedessä kulkeva mobiilirobotti. Viitattu 16.1.2020.

<https://www.unmannedsystemstechnology.com/wp-content/uploads/2020/01/UTEK-4G-controlled-USV.jpg>.

VNC (Virtual Network Computing). 2020. Yhteisön ylläpitämä dokumentti Raspberry Pi yhteisön kotisivustolla. Viitattu 11.11.2019.

<https://www.raspberrypi.org/documentation/remote-access/vnc/README.md>.

What robots do (and don't do) at Amazon fulfillment centers. N.d. Artikkelin Amazon.com yhtiön kotisivuilla. Viitattu 18.1.2020.

<https://www.aboutamazon.com/amazon-fulfillment/our-innovation/what-robots-do-and-dont-do-at-amazon-fulfillment-centers/>.

Zebra. 2020. Zebra Technologiesin kotisivusto. Viitattu 25.2.2020. www.zebra.com.

Zebra UWB Technology. 2018. Zebra sisäpaikannusjärjestelmän datalehti.

Liitteet

Liite 1: Komennot MiR100 sarjaliitännän kautta

Prefix	Syntax	Command	Example explanation	Example	Response
!	R<n>#<value>	Set Register	Set register 101 with value 3.14159	!R101#3.14159	OK: Register set
!	P	Pause Signal	Pause robot	!P	OK: Wait called
!	C	Continue Signal	Un-Pause robot	!C	OK: Continue called
!	GO:<x>,<y>,<theta>	Go to point	Go to point with coordinates (2.4,45.2) 0.29 radians	!GO:2.4,45.2,0.29	OK: Position set
!	GO:<position name>	Go to position	Go to position called "Home"	!GO:Home	OK: Goal position set
!	X	Abort current mission	Abort current mission	!X	OK: Mission aborted
!	MA:<Mission name>	Append Mission to Mission queue	Append Mission "Move to List"	!MA:Move to List	OK: Mission appended
!	MC	Clear Mission Queue	Clear the Mission queue	!MC	OK: Mission queue cleared
?	R<n>	Get Register	Get value in register 101	?R101	OK: R101#3.14159
?	P	Get current position	Get current position (X, Y, theta)	?P	OK: 52.15, 0.81, 1.929
?	S	Get robot status	Get Status (platform state, distance traveled, uptime,battery,mode)	?S	OK: 3, 0.0, 83.87, 25.50, manual
?	L	Get list of available position names	Get list of position names	?L	OK: Home,FAM,Production
?	MQ	Get Missions in Mission Queue	Returns the names of the Missions in the Mission queue	?MQ	OK: Call Elevator, Go to Dock, Dock
?	MA	Get name of active Mission	Returns the name of the active Mission in the Mission queue	?MA	OK: Call Elevator
?	ML	Get name of available Missions	Returns the names of the available Missions	?ML	OK: Call Elevator, Go to Dock, Dock, Production, Open Door

Kuvio 33. MiR sarjaliitännän mahdollistamat käskyt (MiR Serial Interface 2017, 6)

Liite 2: Alustavat Requests-testit

```
import requests
import json

#GET metodin testaus (1. rekisteri)
r = requests.get("http://mir.com/api/v2.0.0/registers/1",
headers={'Authorization': 'Basic
ZGl...k1ZjZmMDU5NjU3NmU0ODJiYjhlNDQ4MDY0MzNmNGNmOTI5NzkyODM0YjAxNA=='})
print(r.text)

#sama toisin
url="http://mir.com/api/v2.0.0/registers/1"
headers={'Authorization': 'Basic
ZGl...k1ZjZmMDU5NjU3NmU0ODJiYjhlNDQ4MDY0MzNmNGNmOTI5NzkyODM0YjAxNA==' }
r = requests.get(url, headers=headers)
print(r.text)

#POST metodin testaus (2. rekisteri)
url="http://mir.com/api/v2.0.0/registers/2"
headers={'Authorization': 'Basic
ZGl...k1ZjZmMDU5NjU3NmU0ODJiYjhlNDQ4MDY0MzNmNGNmOTI5NzkyODM0YjAxNA==',
'Content-type': 'application/json'}
arvo={"value":13}      #vaatii kokonaisluvun, koska
kokonaislukurekisteri
r = requests.post(url, headers=headers, data=json.dumps(arvo))
print(r.text)

#sama toisin
url="http://mir.com/api/v2.0.0/registers/2"
headers={'Authorization': 'Basic
ZGl...k1ZjZmMDU5NjU3NmU0ODJiYjhlNDQ4MDY0MzNmNGNmOTI5NzkyODM0YjAxNA==',
'Content-type': 'application/json'}
arvo={"value":13}      #vaatii kokonaisluvun, koska
kokonaislukurekisteri
r = requests.post(url, headers=headers, json=arvo)#näin import json
turha
print(r.text)
```

Liite 3: IoRT-järjestelmän testaus

```
#Position "transval3" koordinaattien muokkaus
curl -X PUT "http://xxx.xxx.xxx.xxx/api/v2.0.0/positions/5fc4ee0d-48bb-11ea-a05d-94c6911301a9" -H "accept: application/json" -H "Authorization: Basic SW9SVDp1YWl1ZjAwYzAzYTExZDFmM2IwNjNiMzllY2M5OGViYzVjZmNlNjZlYWZjZTllMTA2ODJmNTBkOThkYTQ2NWE4" -H "Accept-Language: en_US" -H "Content-Type: application/json" -d '{"name": "transval3", "pos_x": 90, "pos_y": 24, "orientation": 0, "type_id": 0, "map_id": "58508c4f-5aaa-11e9-8cf7-94c6911301a9", "parent_id": null}'

#Tehtävän lisääminen tehtäväjonoon
curl -X POST "http://xxx.xxx.xxx.xxx/api/v2.0.0/mission_queue" -H "accept: application/json" -H "Authorization: Basic SW9SVDp1YWl1ZjAwYzAzYTExZDFmM2IwNjNiMzllY2M5OGViYzVjZmNlNjZlYWZjZTllMTA2ODJmNTBkOThkYTQ2NWE4" -H "Accept-Language: en_US" -H "Content-Type: application/json" -d '{"mission_id": "dfd01655-4736-11ea-b26d-94c6911301a9", "message": "", "parameters": [], "priority": 0, "fleet_schedule_guid": ""}'

#Robotin tilan hallinta
curl -X PUT "http://xxx.xxx.xxx.xxx/api/v2.0.0/status" -H "accept: application/json" -H "Authorization: Basic SW9SVDp1YWl1ZjAwYzAzYTExZDFmM2IwNjNiMzllY2M5OGViYzVjZmNlNjZlYWZjZTllMTA2ODJmNTBkOThkYTQ2NWE4" -H "Accept-Language: en_US" -H "Content-Type: application/json" -d '{"state_id": 3}'

#Virhetilan kuittaus Trackinno-käyttäjälle
curl -X PUT "http://xxx.xxx.xxx.xxx/api/v2.0.0/status" -H "accept: application/json" -H "Authorization: Basic SW9SVDp1YWl1ZjAwYzAzYTExZDFmM2IwNjNiMzllY2M5OGViYzVjZmNlNjZlYWZjZTllMTA2ODJmNTBkOThkYTQ2NWE4" -H "Accept-Language: en_US" -H "Content-Type: application/json" -d '{"clear_error": true}'
```